UNIVERSIDAD AUTONOMA DE MADRID

Escuela Politécnica Superior

Departamento de Tecnología Electrónica y de las Comunicaciones

AN INNOVATIVE VISION SYSTEM FOR INDUSTRIAL APPLICATIONS

PHD DISSERTATION

Ricardo Ribalda Delgado

Madrid, October 2015

Supervised by
Javier Garrido Salas

PROCEEDING

The Dissertation Committee approves, and recommends for acceptance for the degree of Doctor of Philosophy, the Thesis by Ricardo Ribalda Delgado: AN INNOVATIVE VISION SYSTEM FOR INDUSTRIAL APPLICATIONS.

Dr. Jesús M. González Barahona Chairman	Dr. Guillermo González de Rivera Peces Secretary
Dr. Brian Vinter	Dr. Bjarke Jørgensen

Dr. Gustavo Sutter Capristo

With the grade of:

Madrid, 20th. November 2015

ABSTRACT

Despite the fact that computer vision systems place an important role in our society, its structure does not follow any standard. The implementation of computer vision application require high performance platforms, such as GPUs or FPGAs, and very specialized image sensors. Nowadays, each manufacturer and research lab develops their own vision platform independently without considering any inter-compatibility. This Thesis introduces a new computer vision platform that can be used in a wide spectrum of applications. The characteristics of the platform has been defined after the implementation of three different computer vision applications, based on: SOC, FPGA and GPU respectively. As a result, a new modular platform has been defined with the following interchangeably elements: Sensor, Image Processing Pipeline, Processing Unit, Acceleration unit and Computer Vision Stack. This thesis also presents an FPGA synthetizable algorithm for performing geometric transformations on the fly, with a latency under 90 horizontal lines. All the software elements of this platform have an Open Source licence; over the course of this thesis, more than 200 patches have been contributed and accepted into different Open Source projects like the Linux Kernel, Yocto Project and U-boot, among others, promoting the required ecosystem for the creation of a community around this novel system. The platform has been validated in an industrial product, Qtechnology QT5022, used on diverse industrial applications; demonstrating the great advantages of a generic computer vision system as a platform for reusing elements and comparing results objectively.

RESUMEN

A pesar de que los sistemas de visión por computadora ocupan un puesto predominante en nuestra sociedad, su estructura no sigue ningún estándar. La implementación de aplicaciones de visión requiere de plataformas de alto rendimiento tales como GPUs o FPGAs y el uso de sensores de imagen con características muy distintas a las de la electrónica de consumo. En la actualidad, cada fabricante y equipo de investigación desarrollan sus plataformas de visión de forma independiente y sin ningún tipo de intercompatibilidad. En esta tesis se presenta una nueva plataforma de visión por computador utilizable en un amplio espectro de aplicaciones. Las características de dicha plataforma se han definido tras la implementación de tres aplicaciones de visión, basadas en: SOC, FPGA y GPU, respectivamente. Como resultado, se ha definido una plataforma modular con los siguientes componentes intercambiables: Sensor, procesador de imágenes "al vuelo", unidad de procesado principal, acelerador hardware y pila de software. Asimismo, se presenta un algoritmo para realizar transformaciones geométricas, sintetizable en FPGA y con una latencia de tan solo 90 líneas horizontales. Todos los elementos software de esta plataforma están desarrollados con licencias de Software Libre; durante el trascurso de esta tesis se han contribuido y aceptado más de 200 cambios a distintos proyectos de Software Libre, tales como: Linux, YoctoProject y U-boot, entre otros, promoviendo el ecosistema necesario para la creación de una comunidad alrededor de esta tesis.Tras la implementación de la plataforma en un producto comercial, Qtechnology QT5022, y su uso en varias aplicaciones industriales se ha demostrado que es posible el uso de una plataforma genérica de visión que permita reutilizar elementos y comparar resultados objetivamente.

A mis padres y hermana
A Mariana

ACKNOWLEDGEMENTS

During this (long) journey I have had the pleasure and the chance of being surrounded by an outstanding group of people.

The first person I want to thank is my supervisor, Javier Garrido. He has been undoubtedly the pillar that has supported all the pieces together and has lead them to this final composition. He has been a guide both in my academic and personal life. *Muchas gracias Javier, esta Tesis es tuya*.

The second person I want to thank is my friend and current boss, Kristian Madsen. He has given me freedom to work on my Thesis and trust to execute my ideas in his company. During the last five years I have learn the meaning of the word Engineer. *Tusind tak Kristian*.

Then I want to thank Tom Neubert and the rest of the team at Jüelich (George, Heinz, Tobias, Anne...): I have enjoyed every single day of my internship in your lab where I discover my passion, and now, my work: Embedded Design. *Vielen dank Tom!*

I also want to thank the Open Source community for their help and patience, I am particularly thankful to Hans Verkuil and the rest of linux-media.

It is a pleasure to mention the place that saw and valued my first steps as an engineer and which I remember always with happiness: HCTLAB and all the people there (past and present): Susana, Guillermo, Ángel, Pepe, Fernando, Alberto, Javier Tejedor, Victor, Daniel, Antonio, Nico....

I do not forget the rest of Qtechnology, a lot of the work shown here has been done with their support: Alexandre Desnoyers is the author of most of the PCBs of the QT5022, and a walking encyclopedia for anything with a wire. Finn Nielsen has implemented most of the custom FPGA cores for the QT5022. Jean Claude Thibault has designed the rest of the PCBs and cores. Marianna Buschle has implemented the calibration tools of the device. Dimitrios Katsaros has worked on the

test framework that validate the cameras. Thomas Clausen is the author of the 3D designs. Without their work, this Thesis would be nothing more than another prototype made with a commercial board that has no use outside the Academia.

Other friends have reviewed this text with infinite dedication. Juanjo, you have imprinted a very professional level in this text, thank you very much for all the time expend and I wish you all the best with your new child.

I should mention also two brave colleagues that have made the tough decision to leave Spain to continue with their careers: Bruno and Carmen, thanks for your help mates !

I want to publicly acknowledge my parents for their support during my entire life. *Muchas gracias papás, todo en esta vida os lo debo a vosotros. Nunca podré daros suficientes gracias por los sacrificios que habéis hecho por mi hermana y por mí.*

And of course my wife, for her infinite patience during these years, for being the travel partner of my life and for giving me support and love every single time that I needed it.

I have probably forgot to mention a lot of people, I'm sorry about that, as I said, this has been a very long journey and I am getting too old to even remember my own name :).

Finally, I want to thank you, yes you! whoever you are, for using your time reading this Thesis. I hope that you can enjoy reading it as much as I enjoyed writing it.

CONTENTS

ACRONYMS

ABI Application Binary Interface

AD Analog to Digital Converter

ALU Arithmetic Logic Unit

API Application Program Interface

APU AMD Accelerated Processing Unit

ASIC Application-Specific Integrated Circuit

AXI Advanced eXtensible Interface

BAR Base Address Registers

BIOS Basic Input/Output System

BLOB Binary Large OBject

BRAM Block Random Access Memory

CA Certificate Authority

CCD Charge-coupled device

CDS Correlated Double Sampling

CMOS Complementary Metal–Oxide–Semiconductor

CDMA Central DMA Controller

CPU Computer Processing Unit

CMOS Complementary Metal-Oxide Semiconductor

COTS Commercial Off-The-Shelf

DAC Digital to Analog Converter

DMA Direct Memory Access

DSP Digital Signal Processor

EtherCAT Ethernet for Control Automation Technology

EEPROM Electrically Erasable Programmable Read-Only Memory

FIFO Fist In First Out

FOURCC Four Character Code

FPGA Field Programmable Gate Array

FPN Fixed Pattern Noise

FPS Frames Per Second

GFLOP Giga FLoating-point Operations Per Second

Gib Gigabit 2^{30} bits

GiB Gigabyte 2^{30} bytes

GLORIA Gimbaled Limb Observer for Radiance Imaging of the Atmosphere

GPGPU General Purpouse Graphical Processing Unit

GPIO General Purpouse Input Output

GPU Graphical Processing Unit

HD High Definition

HDR High Dynamic Range

HDL Hardware Description Language

IDE Integrated Development Environment

InGaAs Indium Gallium Arsenide

I2C Inter-Integrated Circuit

IO Input Output

IOCTL Input/Output ConTroL

IP Internet Protocol

IRQ Interrupt

JTAG Joint Test Action Group

Kib Kilobit 2^{10} bits

KiB KiloByte 2^{10} bytes

LED Light Emission Diode

LUT Look Up Table

LVDS Low-voltage differential signaling

MAC Media Access Control

MiB MegaByte 2^{20} bytes

Mib Megabit 2^{20} bits

mmap Memory mapping

NBIS NIST Biometric Image Software

NDA Non-disclosure agreement

NTP Network Time Protocol

OPB On-chip Peripheral Bus

OpenCV Open Source Computer Vision Library

OS Operating System

PC Personal Computer

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

PCIe PCI express

PIO Programmed Input Ouput

PLB Processor Local Bus

PLC Programmable Logic Controller

PTP Precise Time Protocol

PWM Pulse Width Modulation

RAM Random Access Memory

RGB Red Green Blue

SMBUS System Management BUS

SOC Sytem On Chip

SATA Serial AT Attachment

SPI Serial Peripheral Interface

TCP Transmission Control Protocol

UDP User Datagram Protocol

UART Universal Asynchronous Receiver/Transmitter

USB Universial Serial Bus

TV Television

VB2 Video Buffer 2

WOI Window Of Interest

Part I

INTRODUCTION AND STATE OF THE ART

1

INTRODUCTION

Contents

1.1 MOTIVATION

The first Charge-coupled devices (CCDs) were invented by Willard Boyle and George E. Smith on 1969 [WG70]. It took less than a decade for the industry to integrate them into their production chains [Ste79, PCM79, Kos79].

Since then, Computer Vision Systems have play a predominant role on the Industrial Processes; there are multiple logical reasons for this:

- There are processes that cannot be done with Manual Inspection because of the limits of the human perception: We cannot detect artifacts on the invisible bands [BAGM07], or inspect very small objects [RS06].
- Computer Vision Systems give more predictable results and a better error rate than Manual Inspections [MGS98].
- In the long term, a Computer Vision System is financially cheaper than Manual Inspection. A good example are the Optical Graders, with a payback period of less than a year [A/S15].
- Manual Inspection is a very repetitive maneuver than can lead to unhealthy working conditions for the operators [KK79].

Although Computer Vision Systems have evolved in performance and sensing capabilities, both Industry and Academia have felt short to met some of the expectations from the early pioneers of Industrial Computer Vision Systems. Gerald J. Agin in 1980 started his paper Computer Vision Systems for Industrial Inspection and Assembly [Agi80] with this phrase: *Custom-designed computer vision systems are being applied to specific manufacturing tasks. Current development may lead to general-purpose systems for a broad range of industrial applications*. In 2015, 35 years later, that general-purpose system still does not exist to much despair of engineers and researchers in the field.

In 1980, off-the-shelf hardware did not have the capabilities to conform a general-purpose system. The systems of that time were art pieces,

formed by tailor made parts, targeted to an specific application and sold as a black box.

Nowadays, hardware is, by comparison to early 80s, hundreds of time more capable and manufacturers still sell their product as black boxes, even though they are formed by Commercial Off-The-Shelf (COTS) parts. In an interesting semantic paradox, it is worth observing that the word camera origins from the Latin term *Camera obscura* (Dark chamber).

Despite the predicaments on the power of hardware, Computer Vision Systems are installed in almost every production line of each market. What is the challenge or problem then? Is there a need of General-Purpose System? The reality is that the lack of that System is slowing down the overall progress in the field.

This situation is exacerbated by some idiosyncrasies of the research and development environments in the field: on one side there is the Academia, where almost each group has to create their own vision system from scratch, and there is no chance of sharing code and comparing results objectively. Therefore, academic groups develop Galapago Island type of paradigms that are difficult to transfer to Industry. On the other side is the Industry, which is very conservative when it comes to migrating to new systems, as the transfer involved heavy financial costs along with technological uncertainties.

If the society have reached an impressive level of automation without sharing resources, it is easy to dream where could it go with more powerful tools. However, if a Generic Computer Vision System would be such a important milestone, why it hasn't be proposed before?

It would be easy to blame the Industry: if a public standard would exist, the entry barrier to new competitors would be lower, and hence incumbent companies would lose their market share. The reality is that there are a lot of technical challenges on the development of a General Purpose System: there are just too many sensors and processing architectures.

This Thesis implements a novel Computer Vision System that can be used on almost any Industrial Application, regardless of what sensor or architecture they use. This system has been successfully used on a

variety of products, and it has been installed on more than 200 locations around the world.

The major contribution of this System and the reason why it is so versatile is that it is based on a modular structure: The black box has been opened and divided into its main parts. On the hardware side, the components are connected with standard industrial interfaces and on the software side there has been an intense work with the Open Source Community to extend their current interfaces to fulfill the needs of the Industrial Computer Vision.

1.2 DESIGN GOALS

The purpose of this Thesis is the implementation of a generic Computer Vision System for Industrial applications.

The requirement analysis will be done on a very practical way: by the implementation of three custom Computer Vision Systems.

The final system should fulfil the following requisites:

1. Compatibility with a wide range of sensors: From standard sensors, such as CCD or Complementary Metal-Oxide Semiconductor (CMOS) to more rare sensors, such as Indium Gallium Arsenide (InGaAs) or Microbolometer.

2. Support for acceleration hardware like Graphical Processing Units (GPUs) or Field Programmable Gate Arrays (FPGAs).

3. Open Source software stack with a good collection of Image Processing libraries and use of standard interfaces.

All the changes to the Open Source stack should be contributed back to the Open Source community.

The system will be validated against as many applications as possible, preferable in real-life products.

1.3 HISTORY OF THE PROJECT

When a Thesis takes eight years to complete from the day the student obtained his Master of Advanced Studies it deserves a justification:

The initial project of my Thesis was titled: *Identification System based on System-on-Token extensible with certificates* which was granted by the Spanish Ministry of Education and Science.

Finding the right platform for developing that Thesis resulted to be a very difficult challenge. Almost immediately, I realized that the platform developed for my Master of Advanced Studies was not a good candidate, so I moved on to a Sytem On Chip (SOC) that seemed to fulfil my requirements.

Unfortunately, years later I hit a performance wall that I could not overcome; the platform was not powerful enough.

When I was looking for a newer and more powerful platform I realized that each of them required a reimplementation of my application: I had to throw away all the hardware customizations that I did, and most of the software developed.

After my visit to Forschungszentrum Juelich, I learnt two things: First of all, I realized that I have better aptitudes on System Architecture than in Biometry. And then, I discovered that the German Research Centre was facing the same issues as me: They could not find a Computer Vision System that could work on all their systems.

After a very intense internship, I decided to move the focus of my Thesis from Biometry to System Design, and I was lucky enough to be part of a Research Group, HCTLAB, with experience on Generic Robotic Platforms, and there was a lot of ideas from that field that could be useful here.

My new goal was to make a real platform, not a prototype, and soon I discovered that there was too much food on my plate: I needed help and funding, which my University could not provide.

On my journey I met Qtechnology, which had a similar vision and could give me all the support that I needed. I was very lucky to join their team in 2010.

During my job in Qtechnology I realized how difficult it is to make a commercial product, and also how difficult was to complete a Thesis while working on a full-time job.

It took me five years, two completely different designs, and thousands of hours to complete a Computer Vision System that could be used on my original Thesis project. Maybe one day I will make a second thesis with the original title, using the platform developed in this Thesis, I cannot think of a better way to close the circle.

And now, let's go back to the present Thesis.

1.4 STRUCTURE OF THE DOCUMENT

This document is divided in five parts:

1. The first part is dedicated to the Introduction (this chapter) and State of the art (chapter 2).

2. On the second part three different custom Computer Vision Systems are described: one based on a System on Chip (chapter 3), another based on an FPGA (chapter 4) and the final one based on a CPU/GPU (chapter 5). The development of these systems allowed the author to gather the requirements of the final system.

3. On the third part, the proposed Computer Vision System is described: An introduction of the platform is given on chapter 6, then the dataflow is described on chapter 7, to continue with a detailed description of the hardware on chapter 8 and finalize with the software modules on chapter 9.

4. The fourth part is dedicated to the conclusions and future work, there is an English version on chapter 11, and Spanish version on chapter 12.

5. The fifth (and final) part of this document belongs to the appendices, where the contributions to the Kernel (chapter A), U-boot (chapter B) and other Open Source projects (chapter C) are detailed.

The document ends with a list of references.

2

STATE OF THE ART

This chapter gives an overview of the current platforms for Industrial Computer Vision Systems. The state of the art of more specific topics, such as sensing technology or light sources are detailed on the main matter chapters of this document, in order to have self-contained, and easier to read chapters.

Contents

Figure 2.1.: SRI Vision Module

2.1 DEFINITION

There are multiple definitions of Computer Vision, perhaps the most cited is the one by Dana H. Ballard and Christopher M. Brown, in 1989: *"The construction of explicit and meaningful descriptions of physical objects from images"* [BB82].

Reaching to that definition was not an easy job: in 1980, Gerald J. Agin, explicitly said in his paper, Computer Vision Systems for Industrial Inspection and Assembly [Agi80] that *"It is not at all clear what the term "computer vision" means"*.

Figure 2.1 shows one of those firsts Computer Vision System [Bol79].

What it was clear from the very beginning was that this field was going to have a great economical impact for the industry. A.M. Wallace in 1988 projected a revenue of 1200 millions of dolars for 1994: a 2000% increase in 7 years.

Their projections were correct, in 2015 Computer Vision is ubiquitous. It is found in medicine [SC13], army [BMBJ15], surveillance [APO14], food industry [DC+12], electronic manufacturing [Pau12], space research [MMJ+07], road monitoring [SD01] and self-driving cars [Gui11], just to mention a few of the fields where it is present.

2.2 INDUSTRIAL COMPUTER VISION

Industrial Computer Vision is the application of the Computer Vision principles in the Industrial processes.

This is a particularly interesting market because the applications are cost tolerant and more focused in top-performance than any other application [Nie07]. The cost tolerance is due to the fast return of investment in Industrial Computer Vision [Gun00, A/S15]. And the top-performance focus is due to the increased awareness and sophistication of the consumers[BS04, SB00].

The capabilities of an artificial vision system go beyond the limited human perception and deliver much more deterministic results [CAM+11]. A good example of these demands is the corn kernel analyzer from the United States Department of Agricultural Research Service [Pea09], with a throughput of 75 kernels per second.

Aside from the primary applications of Computer Vision, i.e. grading, sizing, defects detection; the objective results from an automatic system can be used as input for a data mining system in order to improving the production of raw materials [web15] (Figure 2.2).

2.3 COMPUTER VISION STAGES

Almost all the Computer Vision Systems follow the same steps: acquisition, pre-processing and perception:

1. Acquisition: In this step an imaging sensor is used to capture an image. Depending on the nature of the scene, different technologies can be used, such as: CCDs [Sun00], CMOSs [AS05], InGaAss [AL10] or Microbolometers [YMB+03]. Chapter 8 gives a more detailed overview of the sensors technologies.

2. Pre-processing: The image is digitally enhanced using image processing algorithms, such as white balance Fixed Pattern Noise (FPN) correction, compensation of geometric defects, et cetera. Some sensors can do some of these corrections on the fly [RVDCJG+08], but

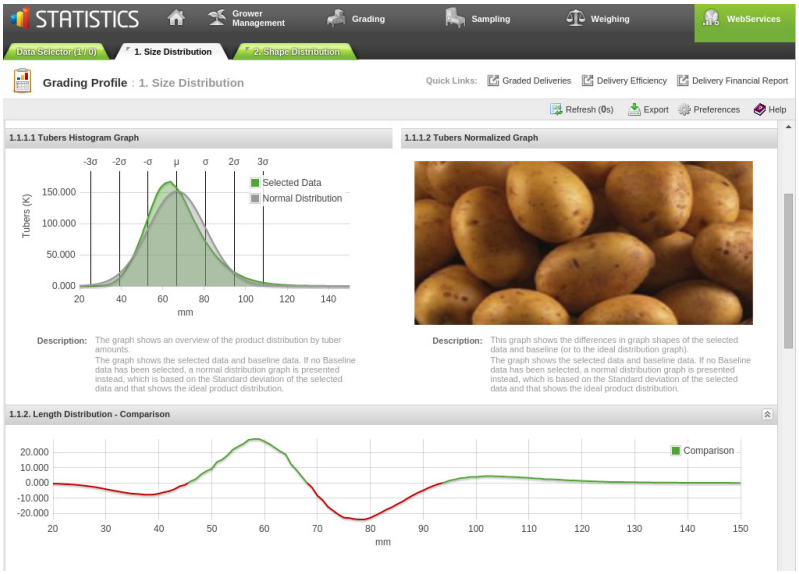


Figure 2.2.: Newtec Web Services Data Mining

this processing is usually done on auxiliary hardware like FPGAs [WK11], or in software [B$^+$00].

3. Perception: The interpretation of a calibrated image that guides the decisions of the computer Vision. This usually follows two steps: Segmentation and Pattern Recognition.

Figure 2.3 from [TCYT12], shows those 3 steps, on the first image a fingerprint is acquired, on the second the image is pre-processed to enhance the minutiae and on the last one, the minutiae are detected.

The image processing algorithms involved in the pre-processing and perception are very demanding and heterogeneous [BCJK05]: On one hand, low-level image processing operations such as interpolation, segmentation and edge enhancement are local, regular and require vast amounts of bandwidth. On the other hand, high-level operations like classification, path planning, and control may be irregular and consume less bandwidth.

Figure 2.3.: Steps in Computer Vision

Sensor's resolution have increased exponentially on the last years. Nowadays it is common to use sensors with tens of megapixels at hundreds of frames per second [HKLP09].

These extreme and heterogeneous requirements make the Computer Vision, one of the biggest challenges of Computer Science.

2.4 COMPUTER VISION COMPUTATION ARCHITECTURES

The high demands of Computer Vision algorithms can only be fulfill with the help of special hardware. These are the most relevant platforms from the literature:

2.4.1 *Digital Signal Processors*

A Digital Signal Processor (DSP) is a specialized processor with an architecture specially designed for Signal Processing. Multiple Multiply-and-Carry operations can be done in parallel with a fraction of the power consumption of a General Purpose Computer Processing Unit (CPU).

They were quite popular on the 90s for Computer Vision and nowadays they still have a niche of systems that require a lot of stability [HRJ+05] or where the price is a deciding factor [BDM+06].

2.4.2 *Field Programmable Gate Arrays*

An FPGA is a chip formed by millions of programmable logic blocks that can be configured to emulate almost any circuit.

In the last years, the number of programmable logic blocks on the FPGAs, and their maximum working speed have increased exponentially.

Thanks to the inclusion of DSPs as programmable logic blocks, they are taking over the market of the DSPs [SB09].

Xilinx, one of the main FPGA manufacturer, have recently released a series of cores and tools for the development of Computer Vision Systems [Vis13].

There are plenty of examples of FPGAs used on Computer Vision [BB14]: detection of potato greening [Sin14], real time stereo vision [WVH97], corn kernel analysis [Pea09] or pedestrian detection [HSH+13], just to cite a few examples.

2.4.3 *Cluster Computing*

Clusters are a group of standard computers working towards the same task.

They are mainly used on off-line image processing algorithms like [MGMEPP+11] or [KGWK96]. The reason for this is the great latency introduced by the load distribution among their nodes.

2.4.4 *General Purpose Graphics Processor Unit*

General Purpouse Graphical Processing Unit (GPGPU) is the use of graphic cards, i.e. GPU, for general purpose problems. The advantage of using GPUs is their great parallelism. They are becoming very popular platform due to its availability.

GPGPU started as a hack, an attempt to exploit the graphic card shaders for something else than image rendering [LM01]. The industry then realized of the great potential of this technology and released new

programming languages for accessing the GPUs hardware on a more structured way [DWL+12].

Soon after the development of CUDA and OpenCL, Computer Vision Systems that made use of the GPUs started to proliferate. Just to cite some examples, GPUs are used today in Aerial photography [Ver11], Face detection [KN12] or even cloud tracking and reconstruction [GGKP08].

Thanks to the popularity and performance of OpenCL, standard Image Processing libraries like OpenCV [Gas14] and video encoders like x264 [MM10] are making use of the computing power of the GPU seamlessly.

2.5 COMPUTER VISION PLATFORMS

A computer Vision Platform is the combination of an Imaging Sensor, a Computation Unit and the software required to process the image flow.

The first platforms were composed by an COTS image sensor connected to a computation unit. The interface between the image sensor and the computation unit could be a proprietary framegrabber [Kum08] or a standard bus like Firewire[RTGM03], USB [MA04] or, more recently, Gigabit Ethernet in the form of GenIcam [Fei15].

On that type of platforms, the functionality of the sensor is limited to the usually poor Application Program Interface (API) provided by the manufacturer. Advanced features such as External Trigger or custom High Dynamic Range (HDR) cannot be done. Also, some of the standard interfaces have latencies of one or more frames.

There are other type of platforms, denominated smart cameras, where the sensor is integrated with the Computation Unit. On the literature the most common smart cameras are based on FPGAs [ST07b, SAW+08, ZWO+11], with some exceptions based on DSPs [BRS04] and SOCs [RRN02].

These systems usually provide better control of the sensor and there is almost no latency on the data capture. The biggest downside is that they behave like a black box: neither the sensor nor the computing unit

can be replaced and despite the fact that they usually run Linux or an embedded version of Windows, the User API is limited to whatever the manufacturer has decided to expose.

There are only three smart cameras on the literature that do not follow the black model concept (Figure 2.4):

- Elphel camera [Filo3]: FPGA based smart camera, where all the components are Open Software and Hardware.
- Standford's Frankencamera [ATP+10]: Multiplatform Linux Based camera with focus in custom trigger timing and accurate flash. Two versions are available, one based on the Nokia N800, and another one based on an Elphel design.
- Carnegie Mellon University CMUCam [RRN02]: Open Source smart-camera designed to provide Computer Vision capabilities to other devices like Arduino or the Raspberry Pi. It has a very limited set of computer vision algorithm, mostly color detection programs.

Unfortunate, none of them uses the current Linux video acquisition system (Video4Linux2) to support their extra capability: instead, they implement their own libraries.

Figure 2.4.: Open SmartCameras
Top Left: Elphel, Bottom Left: CMU Cam, Center Right: Frankencamera

Part II

CUSTOM COMPUTER VISION SYSTEMS

3

EMBEDDED SYSTEM ON CHIP COMPUTER VISION

This chapter describes a Computer Vision System based on a SOC running on a commercial device.

A common biometric algorithm, designed for workstations has been selected as the target algorithm. All the optimizations implemented for fitting the algorithm into the SOC will be described with the impact on the accuracy of the algorithm.

Contents

3.1 INTRODUCTION

In some scenarios, users need to be authenticated by "what they are" instead of "what they own/know". Although biometric verification has many years of life, its implementation in the real world as the substitute for keys/passwords is not so extended.

Formal biometric systems presents a critical problem: biometric patterns cannot be revoked, as the user is born and lives with them. If they are compromised, an impostor could use them to exploit any working system the user has been granted.

Furthermore, there is no biometric technique that can be used for the whole population. Even techniques as mature as fingerprint recognition can't be used universally because of physical problems (like amputations) or sensing problems.

Also, many people don't want to be authenticated by biometrics, they consider it an attack to their privacy.

This chapter introduces a new biometric architecture, based on an Embedded Computer Vision System and denominated System-on-Token, that overcomes all these problems, giving the user full control over his personal data. The implementation discussed on this chapter runs on a commercial System on Chip device, a N800 as shown on Figure 3.1.

3.1.1 *Classical Biometric Systems*

The typical biometric system consists on a huge biometric database and one sensor per application/access. On this scenario, the biometric profiles can be compromised in many points as shown on figure 3.2.

The most obvious attack to gain the biometric patterns from the user, is the compromise of the biometric database. To overcome this problem, the templates on the database should be securized [JNN08], but unfortunately this process affects the accuracy of the system.

Other possible attack vector are the sensors of the system, which could be manipulated or even replaced with malicious devices to steal the personal data.

Figure 3.1.: N800

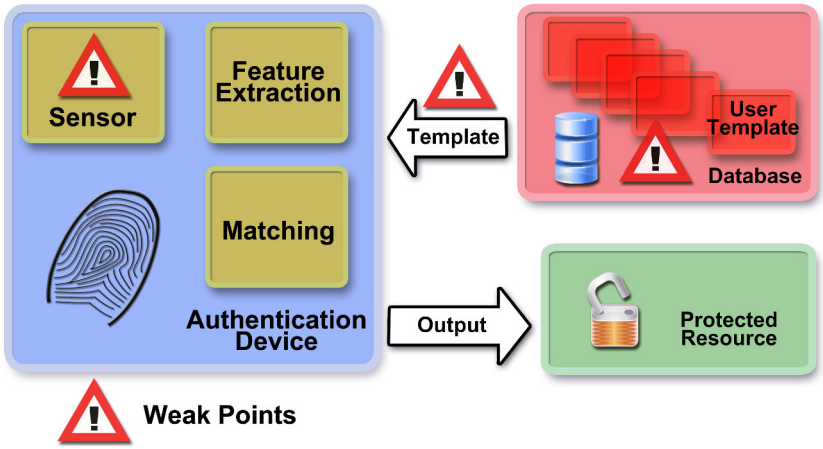


Figure 3.2.: Classical Architecture

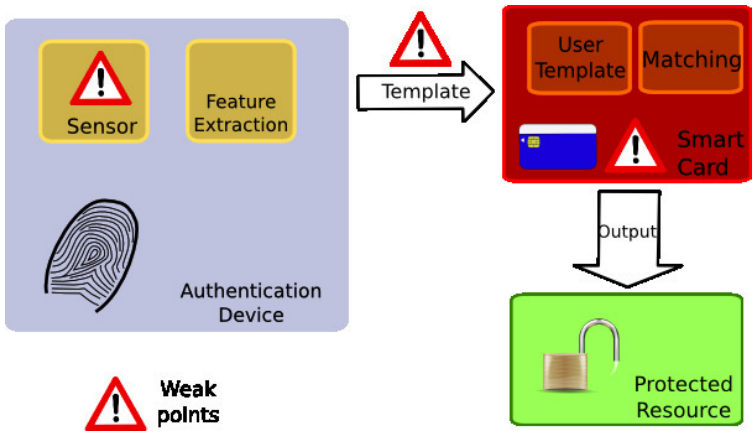


Figure 3.3.: Match-on-Token Architecture

Finally, the biometric patterns could be sniffed or spoofed as they travel through the network,

To overcome the weak points of the classical biometric architecture other works [MGPC03, ISK01] propose a Match-on-Token (Fig 3.3) architecture. On this architecture, the user keeps a token with his biometric data. This architecture removes the attack vectors on the database and the network, but, the user still needs to use a biometric sensor that does not control and could be manipulated.

To overcome this problem, the industry has introduced the Sensor-on-Token architecture, where the user custody the sensor and the biometric profile usually on the form of a smartcard with a fingerprint sensor [Tod02]. This solution removes the implied trust on any sensor but also present some problems: The user has to trust the security of the card.

There is almost no information about what algorithms are working on the card and the quality of them. These systems are developed to be as obscure as the bank smart cards, whose insecurity has been demonstrated earlier [DM07].

Also these cards can only support a small set of biometric modalities due to the strong restrictions on their size.

In 2002 there were at least 15 companies developing smart cards with sensing/matching capabilities [Tod02], 13 years later, none of these companies have overtaken a significant portion of the the banking card market.

Other authors have worked the idea of a System-on-Token [KMW+06, MKS+06] architecture, where the biometric sensor, network interfaces and biometric data lives on an embedded device custodied by the final user. Unfortunately none of these systems can be audited as a whole.

3.1.2 *X.509 digital certificates*

Nowadays, many official formalities can be done remotely thanks to the digital certificates [AF99]. An official organization issues an RSA certificate that allows the user to sign digital transactions anywhere in the world with a computer. This digital signature has the same legal validity as the physical one [GZ03].

The certificates are usually contained in a smart-card or in a password-protected file, this kind of security model follows the classical idea of "what you have" and "what you know".

These certificates follows an asymmetric cryptography model [JK03]. A certificate is composed by a pair of keys, a public ($P()$) and a private one($Q()$). Any document (x) encrypted with the public key ($P(x)$) can only be decrypted ($Q(x)$) using the private key and vice versa($P(Q(x)) = x = Q(P(x))$). The private key is maintained by the user and the public key is released to the public. There is no way of calculating one key using the other ($\nexists f(x) \mid f(Q) = P \; \forall P, Q$).

This cryptography model can be also used for signing documents. To obtain signature (S) of a document (x), the document is firstly hashed ($H()$) and then encrypted with the private key of the user ($S = Q(H(x))$). A signature can be verified using the public key of the signer ($P(S) = H(x)$). Any document signed by the user cannot be repudiated by him / her and the content cannot be modified by a third party.

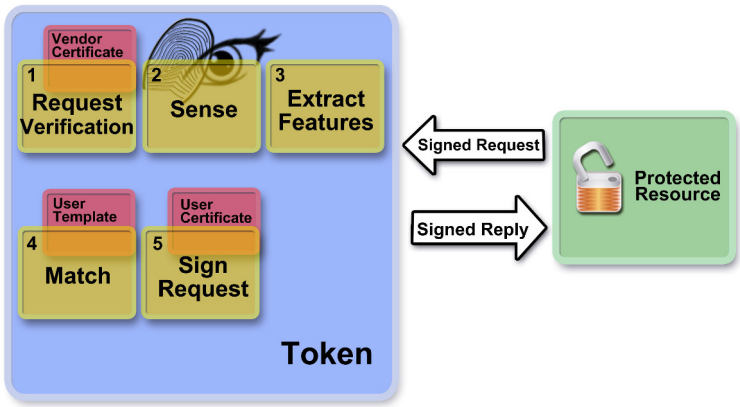


Figure 3.4.: System-on-Token Architecture

3.2 SYSTEM-ON-TOKEN

A Biometric System-on-Token (Figure 3.4) consists on a token carried by the user which can realize all the operations involved in a complete biometric system:

1. Sensing a Biometric Modality.
2. Extract the features.
3. Match the extracted features with other reference features.

The Token should also be able to communicate with the outside world in a secured and authenticated way, providing the user information about the ongoing transactions. The output of the transaction is a signed reply.

System-on-Token is a step further than Match-on-Token and a natural evolution of Sensor-on-Card.

3.2.1 *System on Token Architecture*

Figure 3.5 shows the main parts of a System on Token architecture:

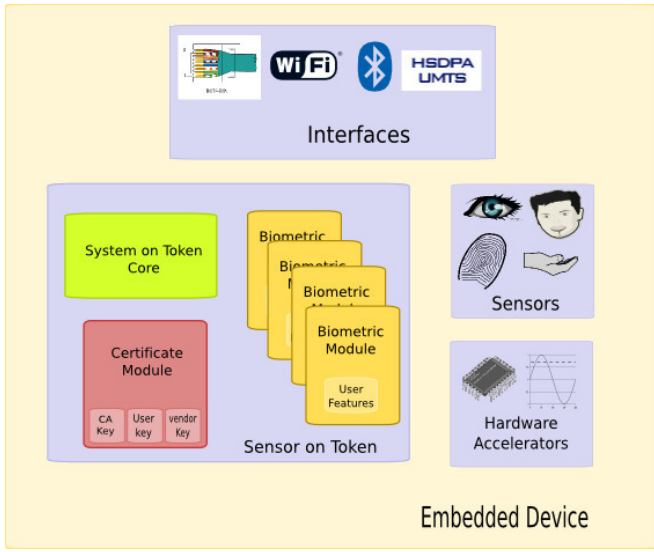


Figure 3.5.: System-on-Token Architecture

- Network interfaces: such as Ethernet, WIFI, Bluetooth, 4G...
- Sensors: Biometric sensors: such as fingerprint scanner, eye scanner, cameras, microphones..
- Hardware Accelerators: used to speed up the Cryptographic algorithms and the imaging algorithms.
- System on Token core: handles the external requests and interacts the other modules.
- Certificate module: Handles the user private and public key as a well as the public key-chain.
- Biometric module: senses, extracts features and matches the biometric patterns.

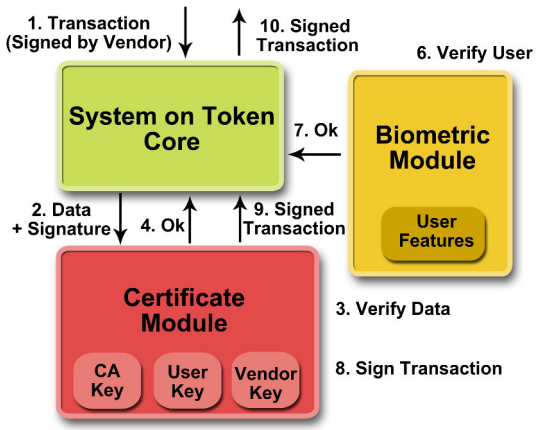


Figure 3.6.: Transaction

3.2.2 *Transactions*

Transactions are the basic operations of the System and have the same legal validity as a physically signed document between the vendor and the user.

The vendor starts the transaction sending a request to the user via any of its interfaces (Fig. 3.6). This transaction is processed by the device which verifies the identity of the vendor and discards the request if the key from the vendor is not trusted.

The user receives all the relevant information of the transactions and if he / she wants to continue with the transaction starts one of the biometric modules.

The biometric module verifies the identity of the user and unlocks the digital key of the user.

The key is used to sing the transaction, which is send back to the vendor.

In this process, no biometric parameters or certificate leaves the device / user.

3.2.2.1 *Other operations*

The device allows adding more vendor keys and other biometric modules, in a secure fashion. All these processes require the active interaction from the user.

3.3 IMPLEMENTATION

In order to implement the System-on-Token architecture on a embedded device there are some important points to take into account.

1. Integration of the sensor in the device: Nowadays, it is common that that manufacturers add biometric sensors to their devices, but this is not a mandatory requirement for the selection of a platform, standard cameras might be used for biometry[LLKK05] or external sensors might be used via Universial Serial Bus (USB) On the Go or Bluetooth interfaces.

2. Integration of the Biometric Module into the system: Biometric algorithms are usually very computationally demanding. The platform should provide the required performance for a smooth user experience.

3. Integration of the Certificate Module in the device: New processors include cryptographic extensions for accelerating common algorithms.

4. Open Source: Drivers / Operating System (OS): The proposed architecture requires that the whole system can be audited, this can only be achieved by the use of Open Source.

3.3.1 *Device Selection*

For this implementation, a Nokia N800 has been selected. This device runs a Linux-based distribution [ST07a] that can be audited and modified.

The hardware is based on a Texas Instruments Omap SOC, composed by a mixed ARM+DSP solution.

This SOC is specifically design for providing a high performance on signal processing algorithms, thanks to its integrated DSP; at the same time as it has a low power profile when it is in standby. This SOC is usually found on mobile phones or internet tablets.

3.3.2 *Biometric Module*

In order to provide a comparable result, the reference implementation for fingerprint processing from the National Institute of Standards and Technology has been used [GWMW01].

The NIST Biometric Image Software (NBIS) software consists on two main parts: Feature extraction (*mindtct*) and feature matching (*bozhort3*). It is completely developed in C/C++ and its source code can be obtained upon request from the NIST.

The initial steps for using the software on the device have been to port the code to ARM, this required taking care of the endian dependent parts of the code.

The performance of the module has been measured with a custom database consisting on 36 fingerprints from different users. These images have been obtained with an optical sensor, Biometrika FX2000, with USB connectivity. This sensor has been previously used on the BIOSECUR-ID database [GFOG+07].

In order to provide a good user experience, the goal of the biometric module was to be able to process a fingerprint under 2 seconds.

3.3.2.1 *Feature Extraction*

The *mindtdt* submodule finds out the minutiae of a fingerprint using an image as the input, it is the most computationally expensive component of the NBIS software.

The ARM port of NBIS, developed for this project, took 5 seconds of processing time per fingerprint.

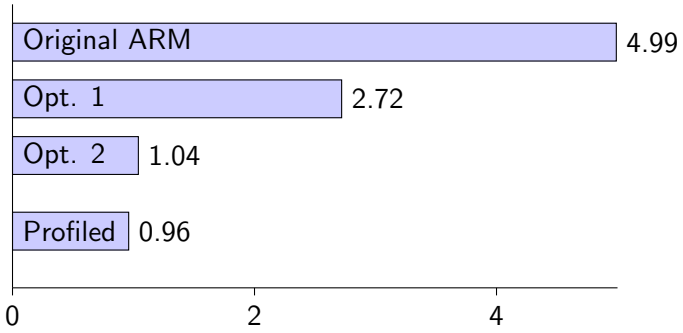


Figure 3.7.: Feature extraction time, in seconds (less is better)
Original is the original version ported to ARM. Opt 1 is the ported version with the standard optimization flags. Opt 2 is the ported version with the best combination of compiler flags. Profiled is Opt 2 with the code optimized after profiling.

An initial optimization was done experimenting with the different compilers flags. This task achieved a speedup of 471%.

A second optimization analyzed the critical paths of the code using a profiler. Two functions were identified and optimized: *dirbinarize* contained divisions that could be substituted with binary operations and *math_first_pair* had a linear search that could be substituted by a hash search. All of these changes did not affect the accuracy of the algorithm and provided another speedup of 8.5%.

Figure 3.7 shows the speedup from the different optimizations.

3.3.2.2 *Matching*

The *bozorth3* module compares two minutiae files obtained with the previous module and gives a score determining their similarity.

Once ported to ARM, and compiled with the best combination of compiler flags it only needs around 0.092 seconds to compare two minutiae files.

Table 1.: Open SSL Speed

Function	Time (in secs)
Hash & Sign 100 bytes	0.13
Hash & Sign 1000 bytes	0.14
Hash & Sign 10000 bytes	0.14
Hash & Sign 100000 bytes	0.15
Hash & Sign 1000000 bytes	0.30
Verify 100 bytes	0.02
Verify 1000 bytes	0.03
Verify 10000 bytes	0.03
Verify 100000 bytes	0.03
Verify 1000000 bytes	0.12
Verify a X.509 certificate	0.07

3.3.3 *Certificate Module*

The Certificates are handled by OpenSSL [VMC02], an open source implementation of the most common Cryptographic systems, including SSL, X509, SMIME and DSA/RSA. This software has been selected due to its broad adoption and the availability of its code.

Different tests has been performed to find out the time needed for validating a certificate, validating a transaction and signing a transaction. All the tests have been realized using an official X.509 Spanish Signature from the FNMT (Spanish Bureau of Engraving and Printing). Results are shown on table 1

3.4 DATAFLOW

This section describes the dataflow of the system from a Computer Vision System point of view.

- The user request via USB a new frame to the sensor.

- The frame is acquired by the sensor and transmitted via USB bulk transfers to the main CPU.
- The CPU launches the Feature Extraction program, producing a list of minutiae.
- The minutiae are matched against the user profile.
- If the profile is properly matched, the certificate is decrypted.
- The transaction is signed and returned to the vendor.
- All the memory is overwritten to avoid cold memory attacks.

3.5 CONCLUSIONS

The proposed system provides a solid alternative for credit cards and a easy way of hardening electronics transactions security with the use of biometry. The computation capabilities of the embedded devices are used to protect a standard X.509 user certificate, using any biometric modality selected by the user.

The biometric parameters of the user and his personal certificate never leave the embedded device reducing the attack vectors. All the system is based on Open Technologies that can be audited by third parties. This platform can be integrated in existing X.509 systems with minor changes.

When considering the use of this Computer Vision System in other applications, there are some downsides to consider. The most obvious problem is the low availability of sensors, the platform is limited by the SOC connectivity, which only has a small selection of sensors focused on consumer electronics.

The different parts of the system cannot be updated, as they are part of a SOC, if more performance or even more memory is required, the whole platforms needs to be replaced, discarding part of the code.

The only accelerator available on the system is the integrated DSP, which lacks a good development environment. Applications need to be rewritten to take advantage of the DSP, which is mainly targeted to multimedia decoding / encoding.

4

FPGA COMPUTER VISION SYSTEM

This chapters describes a custom Computer Vision System with hardware and software preprocessing capabilities based on an FPGA with an embedded PowerPC processor, all running Linux.

This implementation has been used on a commercial 3D fingerprint scanner.

Contents

4.1 INTRODUCTION

Traditional Fingerprint scanners acquire the images either by swiping the finger over a linear sensor or by pressing the finger against a planar sensor. This acquisition presents some problems: The minutiae are obtained from only a small part of the finger, i.e. not nail-to-nail, and the part analyzed is deformed by the pressure against the sensor, producing artifacts on the image.

This is solved with the use of touch-less 3D image sensors [PC09]. These sensors can produce a nail to nail fingerprint image as well as a 3D model of the whole finger (Figure 4.1), all acquired without deforming the finger. Such sensors have very high demands in terms of number of imaging sensor, resolution and computation power that COTS computer vision systems can not fulfil for the following reasons:

1. There is only a reduced selection of sensors available.

2. High resolution cameras with network connectivity usually only deliver compressed images, which, destroys the information from the fingerprint sweat pores, usually referred as level-3 features [JCD07].

3. COTS are non auditable black boxes, this is a major concern in the development of safety equipment.

4. Remote cameras usually lack preprocessing capabilities, which are needed to meet the timing requirements of a user-friendly product.

4.2 FPGAS WITH EMBEDDED CPU CORES

Image processing can be done via hardware or software.

Software processing [FS05], consist on the analysis of an image by a CPU that is usually allocated on the main memory of the device. The high flexibility of the CPU allows an easy development of the algorithm

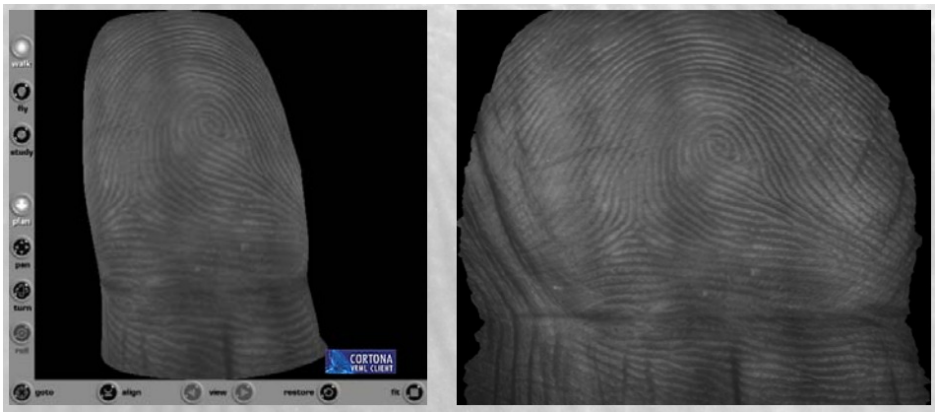


Figure 4.1.: 3D fingerprint

and a very dynamic workflow. On the other hand, the software application requires a lot of time to process linearly the huge amount of data that comes from a sensor.

Hardware processing consist on the use of specialized chips, such as Application-Specific Integrated Circuits (ASICs) or FPGAs [SH06a, LTO05] that process the images on the fly. They provide very high framerates at the cost of a very complex and expensive development process.

The best from both worlds, software and hardware processing, can be achieved via an heterogeneous design. The initial designs that followed this heterogeneous methodology [Gai03] were build around an FPGA with soft CPU cores, such as Microblaze[Xil06] or LEON [TAK06]. The main problem of such systems is the limited performance of the soft-core CPU and the great amount of FPGA resources required by them.

To overcome this problems, the FPGA manufacturers are introducing new families of FPGA with embedded hard-core CPUs, such as the Kintex or Virtex FX from Xilinx. These systems have become the state-of-the-art in high-end development network devices [vdBW05, HE05] and are entering the Computer Vision market.

4.3 ARCHITECTURE DESCRIPTION

In this section, the architecture will be described, including hardware and software elements.

4.3.1 *Hardware*

4.3.1.1 *Sensor*

The noise ratio and resolution imposed by the requirements the final product could only be fulfilled by one Image Sensor: the BCI4-6600 [Vec06], from Vector International, a monochromatic 2208 x 3000 CMOS sensor. It is capable of producing 8, 10 and 12 bit depth images at a 40 MHz pixel rate (aprox 5.5 frames per second). The sensor was interfaced via a proprietary parallel interface.

4.3.1.2 *FPGA*

At the time of developing this system, the best option of FPGA with embedded CPU was the Virtex 4 FX family. Each chip is composed by:

- One or two PowerPC 405 hard-cores.
- One or two TEMACs hard-cores.
- A reconfigurable area, i.e. FPGA fabric.

The PowerPC is used to control the whole device, although it might look like an outdated processor, its use in combination with co-processors can deliver enough performance for the project.

The TEMACs are used for connecting the System to a Gigabit Ethernet network, giving a maximum theoretical bandwidth of 1000 Megabit 2^{20} bits (Mib)/sec, enough for high resolution video at high framerates and RAW format.

The FPGA fabric is used for buses, glue logic to the camera and other cores such as serial port, timers, etc.

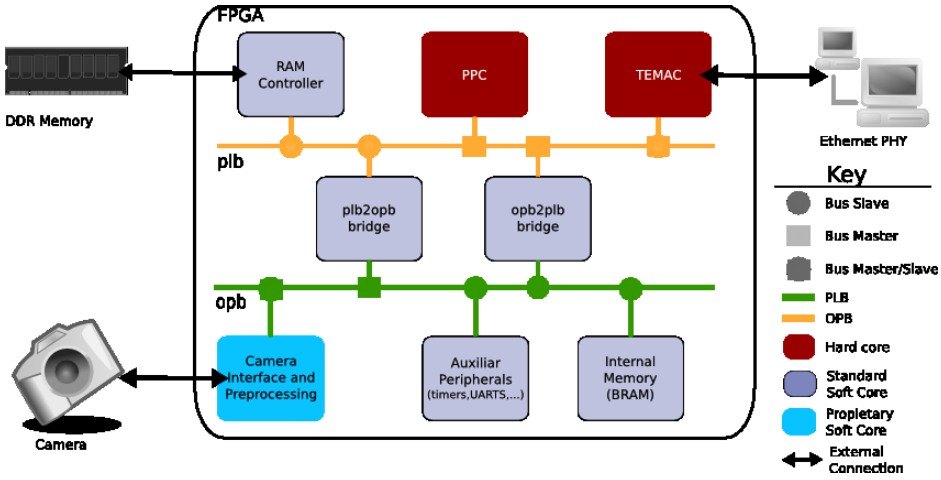


Figure 4.2.: FPGA design

4.3.1.3 *Buses & Specific Peripherals*

The PowerPC is accompanied by a set of cores used to access the external peripherals and pre-process the image. The cores are accessible via two different buses: Processor Local Bus (PLB) and On-chip Peripheral Bus (OPB). PLB is designed for high speed peripherals such as the TEMAC or the Direct Memory Access (DMA) engines and OPB is a much simpler bus used for low speed devices such as serial ports or General Purpouse Input Outputs (GPIOs).

On this design, the Xilinx cores have been used wherever possible. Custom cores are only used for accessing the sensor and for image processing algorithm based on convolution matrices [JGB04].

Peripherals are mapped into the PowerPC memory which access their registers via *iowrite() / ioread()* interface. The frames are copied to the memory via DMA.

The final FPGA design is shown is shown in the figure 4.2.

4.3.1.4 *External Devices*

Due to the limited memory resources on the FPGA it needs to be complemented with an external memory.

A non-volatile memory is also used to keep the FPGA configuration and other information such as manufacture data and the initial bootloader.

LEDs and serial ports are used for debugging purposes.

4.3.2 *Software*

4.3.2.1 *Operating System*

Although the PowerPC can run a bare metal application, it can take great advantage of the multiple services and abstractions already implemented on an OS, such as multitasking, memory management or network stack, among others.

Linux has been chosen as the OS for the camera, due to its code availability and great support from its community. Linux supports the PowerPC hard cores found on the Virtex 4 FX FPGAs and most of the Xilinx cores from its version 2.6 [Tor].

In this particular project, the only required change to the standard Linux version, has been a new module that adds support for the Image acquisition core. The module provides a custom *ioctl* and *mmap* interface for accessing the sensor. A userland library has also been developed for interfacing the module.

4.3.2.2 *Linux Distribution*

Instead of choosing an already available distribution, the limited amount of Flash memory has favored the development of a tailor made distribution based on Busybox[And08].

Due to the biometric nature of the project, special care has been taken to reduce the possible attack vectors to the system, this has been achieved by auditing the system and removing any unused library and service.

4.3.2.3 *Bootloader*

Before the kernel can be started, it needs to be loaded into the memory and the main memory needs to be initialized, this is done with a bootloader.

In the last years, U-boot is becoming the de-facto standard bootloader, with support for multiple boards and architectures and a lot of extra functionality such as network stack, memory test or even tools for bringing up the board.

As a contribution of this Thesis, U-boot has been ported to the embedded PowerPC available on the Virtex 4 FX and Virtex 5 FX. This change has been included in the official branch of U-boot.

4.3.2.4 *Data Flow*

Hardware and software work closely during image extraction and preprocessing. The system works following these steps:

1. The application on the PowerPC starts the image acquisition with an Input/Output ConTroL (IOCTL) to the custom module.
2. The custom module, via a register write to the acquisition core, starts exposing an image.
3. Once the image is exposed, the acquisition core reads the image from the sensor, via a parallel interface and process it on the fly.
4. The data is pushed to the main memory with a DMA transaction.
5. When the DMA transfer has finished, a interrupt is sent to the PowerPC.
6. The custom module services the interrupts and notifies the application about the new frame.
7. The application memory maps the new frame and process it with the CPU.
8. The result image data is sent through the Ethernet interface via DMA.

4.4 VERIFICATION AND RESULTS

4.4.1 *Hardware Implementation*

On the current implementation, it has been decided to use as many COTS modules as possible to reduce the engineering and verification time.

The hardware design is shown on the Figure 4.3. On that image it can be observed the different elements of the system. From left to right.

- Virtex FX12 Mini-Module, from Avnet.
- Custom Mother board, and Firewire interface from Vector (not used on the final design).
- BCI4 readout board, from Vector.
- BCI4-6600 Sensor, from Vector.
- Fixed C-MOUNT lens, from Schneider optics.

The Virtex FX12 Mini-Module by Avnet is an affordable development board, less 200$ per unit, with a reduced form factor, 30 mm x 65.5 mm, and 76 user available I/O pins.

The custom motherboard serves as power supply for the FPGA board and the BCI4 sensor as well as populating the debug connectors for the FPGA.

The whole design, including options has a price under the 1000$. More than reasonable for a low volume, custom design.

4.4.2 *Communication Speed*

Some experiments have been performed in order to measure the transfer speed between the platform and an external device.

For each iteration, 20 images with a size 8 MiB have been transmitted from the camera to a high-end computer. These different methodologies have been used for the data transfer:

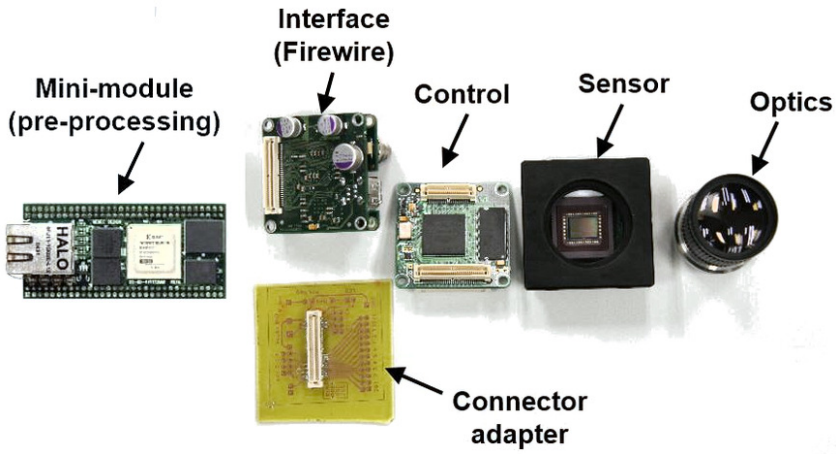


Figure 4.3.: Hardware used

1. UDP vs. TCP: Measure the overhead of the different network protocols
2. ZeroCopy: Measure the overhead of copying the frames to a new memory location, versus sending the frames in place.
3. DMA: Speed gain for using DMA on the network interface.
4. MTU: Implications of the datagram size on the transfer.
5. Link Speed: Speed gain for using a Gigabit network.

Results from these experiments are shown in Table 2.

4.4.3 *Image processing*

The implemented architecture can process the sensor images with hardware and software.

Hardware algorithms are developed in Hardware Description Language (HDL) using the standard Xilinx tools. On the final product, white balance and a convolution kernel were implemented. Figure 4.4 shows

Protocol	ZeroCopy	MTU	DMA	Gigabit	Mbps
TCP	Read	1500	No	No	32
TCP	Read	1500	Yes	No	40
TCP	Mmap	8500	No	Yes	40
TCP	Read	8500	No	Yes	40
UDP	Read	1500	No	Yes	56
UDP	Mmap	1500	No	Yes	64
UDP	Read	8500	No	Yes	80
TCP	Read	1500	Yes	Yes	88
UDP	Mmap	8500	No	Yes	104
UDP	Read	1500	Yes	Yes	192
TCP	Read	8500	Yes	Yes	200
UDP	Read	8500	Yes	Yes	256

Table 2.: Communication Speed Experiments Results

a simple vertical edge detection implemented in hardware with a convolution.

Software algorithms run on the Linux kernel, processed by the PowerPC. The only image processing library provided by the system is used to interface the custom kernel module. Figure 4.4 shows a simple pattern recognition algorithm trained for looking for the letter 'F'.

4.5 CONCLUSIONS

This chapter shows a custom made computer vision system implemented on an FPGA with an embedded PowerPC processor. This system is capable of acquiring and processing up to 5 frames (6.6 Megapixels) per second with hardware and software processing capabilities.

This platform has been successfully used on a commercial, state-of-the-art product.

Although most of the parts of the system are COTS, the development of this platform required almost a whole academic year. Most of this

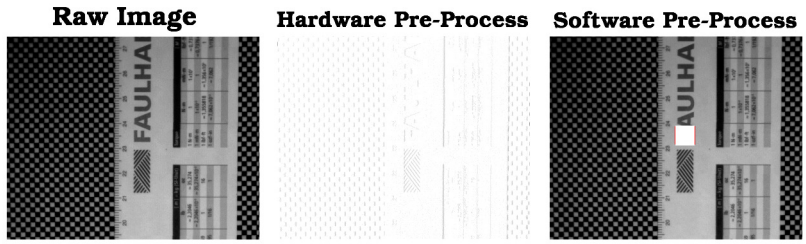


Figure 4.4.: Pre-Processing Algorithms

work cannot be easily reused because the whole system has been optimized for the selected sensor.

In order to achieve real-time performance, most of the processing needs to be done on the FPGA fabric, developed using HDL languages, a tedious process with a very slow development and validation cycle.

5

GPU COMPUTER VISION SYSTEM

This chapter describes Computer vision System based on a standard Personal Computer (PC) architecture accelerated with a GPU.

This system has been used on Gimbaled Limb Observer for Radiance Imaging of the Atmosphere (GLORIA), a Fourier transform spectrometer that is capable of operating on various high-altitude research aircraft.

Contents

5.1 INTRODUCTION

High altitude spectrometry is one of the best tools for measuring air quality and forecasting weather. The traditional instruments are composed by a 1-dimensional spectrometer, which provides a very limited measurement from the sky.

There is a combined effort from multiple German research institutes to create a high altitude bidimensional spectrometer, with a spatial near infrared sensor, denominated GLORIA (Figure 5.1).

Among these institutes, Juelich Forschungszentrum is in charge of implementing the acquisition and first levels of processing of the device. The original implementation required 3 hours and 30 minutes of processing time for a interferogram that was captured in 90 seconds, which disabled the use of the instrument in weather forecast.

Thanks to the work developed on the previous chapter for U-boot, the author got in contact with Juelich Forschungszentrum. Once they were aware of his interest in Computer Vision System, a 3 month internship on their institute was arranged, where the main goal was to accelerate the first level of processing of the device.

Although the acquisition sensor could not be modified, there was complete freedom to use any platform compatible with a PC to achieve the timing goal. As a result of this work, the first level of processing was accelerated more than 200 times.

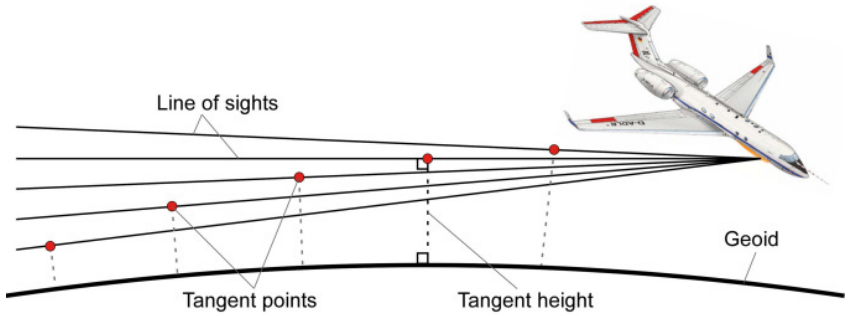


Figure 5.1.: High Altitude plane using GLORIA for acquiring interferograms

5.2 SPECTROGRAM ACQUISITION

Figure 5.2 shows a detailed figure of GLORIA mounted on a high altitude aircraft.

The instrument is composed by two sensors, an infrared image sensor and a laser reference system.

The optical path difference of the interferometer is measured with a laser reference system. A diode laser signal with a wavelength of about 646 nanometers, is coupled into the interferometer, and the rising zero crossings of the laser interferogram are detected and marked with a timestamp. These timestamps are stored as laser data files. The timestamps for both the infrared and the laser signal are generated by a 80 MHz clock within the interferometer electronics. They are given in integer time ticks with one tick corresponding to 12.5 ns.

The infrared image sensor, takes images at a constant frame rate of 2665 frames per second, with 128x128 16 bits pixels.

Each measurement consists of one cuboid containing the infrared signals sampled at a constant frequency (.bin file), and a laser file containing the timestamps of the rising zero crossings of the laser reference system (.lxa file).

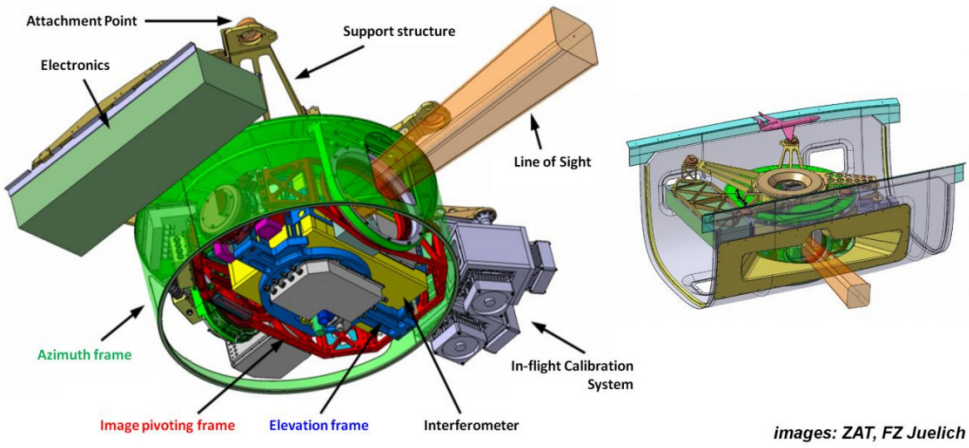


Figure 5.2.: HALO instrument

The core of the level 0 processing is the resampling of the interferograms on a space-equidistant grid using the information from the laser file and the infrared signals. The resampling comprises the spectral calibration by taking the correct laser wavelength and the off-axis angle for each pixel into account. Laser wavelength and off-axis angle are determined from in-flight measurements

While the level 0 processing is done, other calibration operations needs to be taken into account such as quality control of the data, linear phase correction, nonlinearity correction and spectral calibration.

5.3 PROCESSOR OPTIMISATION

The data processing described on the previous section has to be performed for each individual pixel, i.e. several thousand times per measurement. This requires some optimisation in the processing strategy. High time consuming operations like the interferogram resampling have to be optimised.

Since the processing algorithm is still under development, a two-track strategy is followed: At first, the processing is programmed in IDL (Interactive Data Language), using the heritage of the processor that was developed for previous instruments [FVMS+04, WOG+12]. The IDL processor with the character of a prototype allows simple diagnostics and visualisation, and it is easy to modify for testing purposes. Later on, when the IDL prototype is validated, a second implementation in the more efficient computer language C is developed.

On the first level of processing, each pixel is resampled, corrected for nonlinearity and spectral calibrated, taking into account the external computed delay and phase correction.

5.3.1 *Motivation*

Due to GLORIA's imaging capabilities and high-speed data acquisition system, the instrument produces a rapid data output. During the initial campaigns, the detector was configured to run at a sampling frequency of 6128 Hz using 48x128 of its pixels. The signal of each pixel per sample is represented by a 16-bit-wide integer value. With this setup, the detector's data rate totals at 71.8 MegaByte 2^{20} bytes (MiB)/s. Together with the timestamps obtained from the laser interferometer, whose data rate is negligible in comparison. The measurements are saved into two files: .bin and .lxa,

With this amount of data, the processor has to be optimised in order to approach or possibly even exceed the data acquisition speed. This is especially important for possible future long-duration balloon or satellite missions where it is not possible to store all the raw data and on-board processing may become important for data reduction.

5.3.2 *Description of the Operational L0 Processor*

The algorithm used to transfer the interferogram from the time domain into the spatial domain is an approximate Whittaker-Shannon interpolation as proposed by [Bra96]. The spatially equidistant data from the

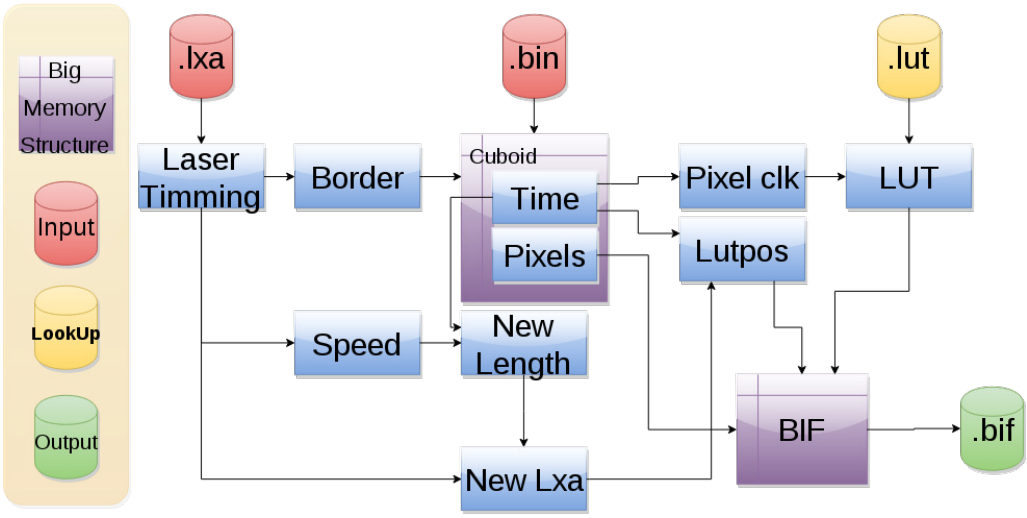


Figure 5.3.: Level 0 processing

laser interferometer is used as a reference axis. This transfer from the temporal into the spatial domain is necessary for the subsequent (L1/L2) processing stages. An overview of the level 0 processing can be seen on figure 5.3.

Mathematically, the interpolation corresponds to a convolution of each pixel's interferogram data with a modified sinc function. This is implemented in the L0 processor using a set of finite-width convolution kernels. The kernels are pre-tabulated to avoid computationally expensive runtime calculations of the cardinal sine values. In order to reduce the size of the whole lookup table, the kernel values are stored in an integer (i.e. fixed point) representation.

For each sample on the target axis, the correct kernel is chosen from the table depending on the corresponding laser timestamps. To avoid nonlinear memory access, a data structure has been specifically designed which maps each point of the pre-defined output abscissa to the correct input data and convolution kernel.

Due to the effect of cache misses, a prerequisite for the efficient convolution of the data is that the interferograms be contiguous in memory.

Unfortunately, due to the nature of the data acquisition, the input is structured in a transposed form. The transposition of the input data is therefore the first processing step, followed by the interpolation setup and, finally, the Brault convolution itself.

To ensure performance and compatibility, the L0 processor is written in the C programming language. The reimplementation in C and the lookup table approach alone, without parallelisation, already resulted in a speedup of two orders of magnitude compared with the original prototype.

5.3.3 *Performance*

The runtime of the L0 processor is, apart from file input/output, dominated by two calculations. The first is the transposition of the measured image, the second the Brault convolution itself. Several optimisations have been implemented to accelerate these computations. Fortunately, both are by their nature very well suited for shared-memory parallelisation.

Further potential for optimisation was discovered by analysing the assembly code generated by the C compiler. Using low-level instructions provided by the SSE instruction set, the vector registers of modern CPUs could be used for optimised reimplementations of both the resampling and the cuboid transposition.

Both approaches, parallelisation and vectorisation, were combined and resulted in a high-performance system able to match and exceed the data rate of the instrument. All following runtime measurements have been performed using a single GLORIA chemistry mode measurement. The cuboid file in question is about 945 MiB in size and took 12.9 seconds to record. The CPU of the benchmark computer was an AMD Opteron 6128 with 8 cores clocked at 1800 MHz. In order to eliminate the impact of disk I/O, the relevant input files have been read from a RAM-based tmpfs file system, which is also used for the output.

Figures 5.4 and 5.5 show the runtimes and speedup factors for the Brault convolution, the cuboid transposition and the total processing

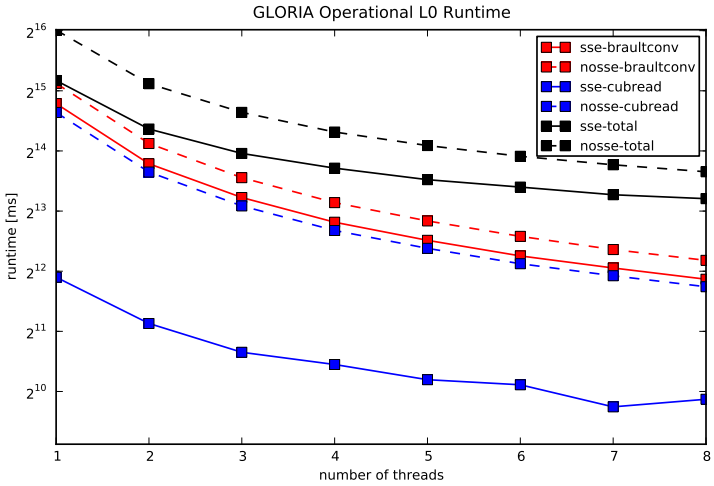


Figure 5.4.: Runtimes with and without OpenMP parallelisation

time for the non-vectorised and the vectorised implementations. The non-vectorised version takes about 56.4 seconds to run without parallelisation, which is reduced to 12.4 seconds using 8 concurrent threads. Note that this already slightly exceeds the real-time speed target. With the SSE optimisations included, these timings reduce further to 36.9 seconds (single-thread) and 9.5 seconds (8 threads), respectively.

The most efficiently parallelised algorithm is by far the Brault convolution, whose speedup scales almost linearly with the number of threads employed. The cuboid transposition involves a larger non-parallel overhead, making it scale less ideally. The use of SSE instructions mitigates the benefits parallelisation only slightly, so that using both in conjunction is very effective.

The cuboid transposition benefits most from vectorisation with relative speedups larger than 4 in single-thread and still larger than 3 in 8-thread mode. The speedup is expected to become less prominent when more threads are used concurrently because memory bandwidth

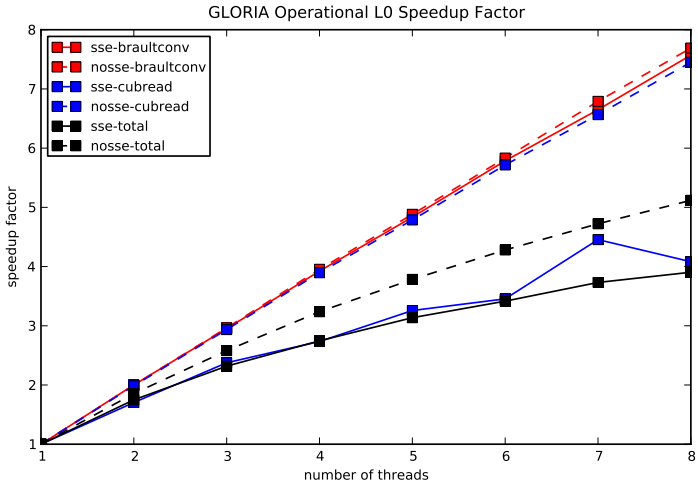


Figure 5.5.: Speedup factor with increasing level of parallelisation

remains limited. In contrast, the speedup for the Brault convolution is almost constant at 25%, while the processor as a whole runs between 31% and 54% faster with SSE enabled.

5.4 GPU ACCELERATION

Although the optimized C implementation of the level 0 was already capable of processing the interferograms on the fly, the nature of the data made it a good candidate for an implementation with GPUs, specifically NVIDIA Tesla boards, already own by Juelich Forschungszentrum.

As previously described, the convolution is a good candidate for GPU optimization: it is executed identically for all the pixels, there is almost no flow control instructions, and it is mainly composed by arithmetic operations.

On the CPU implementation, once the lookup table values are loaded, the resampling of the data is done with two nested loops:

```
for (i=0;i<n_pixels;i++){
        p_value=&orig_ifg[lutpos[i].cuboid_position];
        p_kernel=lut->gridx_point[lutpos[i].lutpos];
        //Convolution
        new_value=0;
        for(j=0;j<KERNEL_SIZE;j++){
                new_value+=(p_value[j].value)*
                    (p_kernel[j].value+1);
        }
```

This can be replaced with 4096 GPU threads, each one processing 4 different pixels in parallel:

```
uint32_t pixel=(threadIdx.x+
            (blockDim.x*blockIdx.x))<<2;
my_pointer_cuboid=pointer_cuboid+pixel;
//#pragma unroll 17
for(j=0;j<KERNEL_SIZE;j++){
        //Get data
        p_indata=(ushort4 *)
            &(ifgin[my_pointer_cuboid].value);
        indata[j]=*p_indata;
        lut_data=sharedlut[j];
        lut_data++;

        //Multiply data
        res.x+=indata[j].x*lut_data;
        res.y+=indata[j].y*lut_data;
        res.z+=indata[j].z*lut_data;
        res.w+=indata[j].w*lut_data;

        //Next data
        my_pointer_cuboid+=n_pixels;
}
```

The initial measurements showed a speedup of only 8% to the CPU implementations, but further analysis revealed that the GPU was only computing 33% of the time, which could results into a optimization of 69.3% with the right pipelining.

5.5 CONCLUSION

The implemented Computer Vision System has successfully met all the goals for the project, this is, delivering processed interferograms in real time. This system has been successfully used in high altitude missions for two seasons on two different aircraft with satisfactory results.

The combination of CPUs and GPUs on the processing pipeline can deliver an acceleration of up to 69%, i.e. 3.2 times faster, than a CPU only architecture. The simplicity of the GPU tools allow a simple port of the algorithms implemented in C to the GPU. In this case, the porting effort has been the identification of the inner loop and the addition of pragmas.

Because the CPU implementation already met the initial goals of the project and does not require extra hardware, it has been the implementation used in the missions so far.

With some perspective, there is also space for improvements on this computer vision system: First of all, the data from the sensor could be preprocessed with an FPGA to ease the level 0 processing. Then, the file interface between the acquisition hardware and the level 0 software could be replaced with a more efficient memory mapping interface, reducing latencies and memory copies. Finally, the processing unit could have some OpenCL capabilities for parallelizing the processing without an external card.

Part III

MODULAR COMPUTER VISION SYSTEM: QT5022

6

MODULAR COMPUTER VISION SYSTEM

In the previous sections of this Thesis, three different Computer Vision Applications have been presented. Each of them required a custom made acquisition and processing system, which involved a large engineering effort.

The main objective of this Thesis is to construct a novel Computer Vision System that can be efficiently used in almost any application, removing the burden of implementing a new acquisition and processing system for each project. The design of this Computer Vision system is a result of the lessons learned in the development of the early described platforms. This chapter serves as an introduction of this Computer Vision System, including a high level description of its architecture.

Contents

6.1 COMPUTER VISION SYSTEMS: DESIGN PATTERN

After developing multiple Computer Vision Applications, it has been identified a common design pattern used on their development:

i There is an initial research stage, where the nature of the problem is understood and the requirements of the system are discovered. During this stage, the solution space is explored through experiments with multiple acquisition system. Each acquisition system is programmed in different ways and the results require normalization before they can be compared. At this point, the computing requirements and acquisition technology are estimated.

ii Once the acquisition technology is defined, the imaging sensor system is identified. Sensors can be classified in different ways; chapter 8 of this Thesis gives a detailed overview on imaging sensor technologies.

iii Once an adequate sensor is selected, the market is explored to find a Computer Vision System that integrates the chosen sensor and provides the required computing power. On the event that the right system cannot be found, it should be developed from scratch, adding a new order of magnitude to the complexity of the project.

iv The computer vision system will provide a set of libraries for acquiring and processing the images. They conform the Image Processing Stack that the designer will use to build the application.

This methodology yields some challenges:

- It has a linear flow, where each step has full dependency to the previous one. A reiteration of the development cycle may invalidate the whole design.
- Algorithms developed for one Computer Vision System needs to be adapted, or even re-engineered from scratch to use them to other system.

- The operation of a particular Computer Vision System involves a series of skills that are not applicable to any other system, such as device Calibration or use of a custom Integrated Development Environments (IDEs).
- Results from different systems cannot be directly compared.

On the hypothesis of the existence of a configurable generic Computer Vision System that could be used for any application, the previous design pattern can be simplified to:

i On the research stage, the system designer acquires images with different configurations of the same Computer Vision System. The same acquisition software can be used in each test and the results are directly comparable.

ii Once the acquisition and computation requirements are determined, the Computer Vision System is configured with the appropriate sensor and acceleration hardware.

iii An standard Image Processing Stack is used to develop the application. Code can be reused from previous projects.

Such a design pattern overcomes the previous problems and let the user of the System focus on the development of the Computer Vision Application.

6.2 MODULAR COMPUTER VISION SYSTEM

The previous section identifies the advantages of a modular Computer Vision System. This section identifies its main components.

Before the generic Computer Vision is defined, it is important to identify the basic elements of a Computer Vision System. This list is has been build through the experience acquired from the development of the applications described on the previous part of this Thesis.

- Imaging Sensor: Converts the light into a quantifiable electrical signal.

- Image Processing Pipeline: Converts the quantifiable electrical signal from the sensor into an array of data. This element has preprocessing capabilities.

- Processing Unit: It is the element that drives the whole system. It is composed by a CPU running the user application with the drivers needed to control the Image Processing Pipeline and the (optional) Acceleration Unit.

- Acceleration Unit (optional): Massive computing unit capable of of handle arrays of data in a parallel form.

- Computer Vision Stack: Group of libraries that can acquire images and implements a great number of image processing and computer vision algorithms. It serves as the interface to the system for the user application.

In a Standard Computer Vision System, as envisioned by the Industry and the Research community, all these elements are part of a black box that is interfaced by the user. The scope of a Standard Computer Vision System is constrained by the specifications of its immutable components, and is not ready for the future applications.

This Thesis proposes a modular Computer Vision System, composed by the blocks identified earlier (Imaging Sensor, Image Processing Pipeline, Processing Unit, Acceleration Unit and Computer Vision Stack). Each block has a defined interface and can be replace with any other module that follows that interface. Figure 6.1 gives a general view of the proposed Computer Vision System. On the figure, the elements are represented with boxes, and the interfaces with arrows.

The concept of a modular system is not new, the revolution of the Personal Computers was achieved through modularity and Mobile phones are initiating this trend through the project Ara [Mur, YY15]; this Thesis proposes applying modularity to Computer Vision Systems. This new approach to Computer Vision System, composed by an heterogeneous modular system is one of the keys contributions of this Thesis.

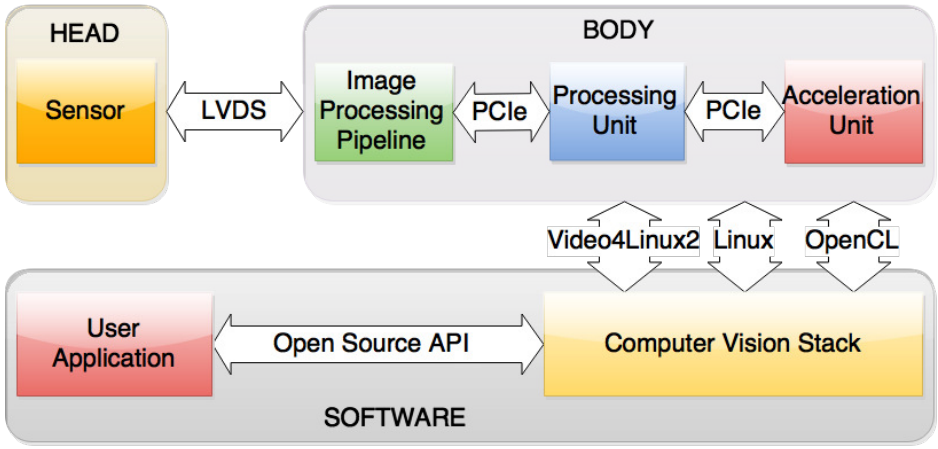


Figure 6.1.: Proposed Computer Vision System: Elements and interfaces

6.3 INTERFACES ON A MODULAR ARCHITECTURE

The previous section has identified the advantages of a modular Computer Vision System. This section analyzes the properties of the interfaces between the modules of the system.

An interface is a description of how an element interact with other elements or the user. The key properties of a good interface are: wide adoption, simple implementation and good performance [Bloo6]. These properties can be found on standard industrial interfaces such as Low-voltage differential signaling (LVDS) and PCI express (PCIe), or in software libraries like OpenCL and the Kernel API.

During the development of this Thesis, it has been identified that the Linux Kernel lacked some functionality for supporting Industrial Computer Vision Systems. This has been workarounded, by the other Computer Vision Systems in the literature, with the creation of custom interfaces. In contrast, this Thesis takes the approach of working in collaboration with the Community behind the Kernel API to extend its functionality. This is another of the key contributions of this Thesis in

the area of Open Source, and future Computer Vision Systems that use Linux will benefit from this work.

6.4 QT5022: A MODULAR COMPUTER VISION SYSTEM

The previous sections describe the elements of a Generic Computer Vision System and the properties of its interfaces. This section provides a high level view of an implementation of such System.

Qtechnology is a Danish Company specialised in Computer Vision applied to the Food Industry. Their first product was an FPGA-based smart camera with a similar structure to the described in this this Thesis, but developed independently. Both systems failed to scale up to the increasing performance requirements of the state of the art Computer Vision Algorithms used in the Food Industry. And both systems failed to support a rich sensor portfolio. Also, they lacked a rich Computer Vision Stack.

As an attempt to meet the expectations of their clients, Qtechnology established a collaboration with the author of this Thesis in order to identify the requirements of a high performance smart camera and implement it. In this collaboration, Qtechnology contributed their knowledge of the industrial environment and their expertise in developing real-life products. The author of this Thesis contributed the experience on high-performance computer vision gathered from the applications described on the first part of this Thesis and the experience with generic robotic platforms. This collaboration ended up being the perfect scenario for implementing the concept of a modular Computer Vision System in a new product named QT5022.

QT5022 follows the modular design described on the previous sections. The current implementation has a total of 19 different sensors, 2 different Image Processing Units, a single Main Processing Unit, 2 different Acceleration Units and a rich Computer Vision Stack composed by over 11000 packages.

The elements of the QT5022 are located in two different mechanical parts: The Body and The head. The mechanical division was made for

Figure 6.2.: Qt5022 Computer Vision System

two reasons: Isolate thermally the sensor from the rest of the elements and an easy exchange of sensors for the final user.

Figure 6.2 shows a photo of a QT5022 with a CMOS CMV2000 sensor and a C-Mount lens.

6.4.1 *Head*

The head is the mechanical group of the Computer Vision that holds the sensor. Figure 6.3 shows a mechanical assembly of a Head unit with the following parts (left to right): Lens mount, protection glass, sensor holder, sensor (CMOSIS CMV4000) and the readout Printed Circuit Board (PCB). The interface to the Body is a PCB cartridge with 16 LVDS lines.

The sensor is mounted on a thick metal frame that serves as a passive heat sink and gives robustness to the design. The Head can be easily replaced by the user removing 4 screws.

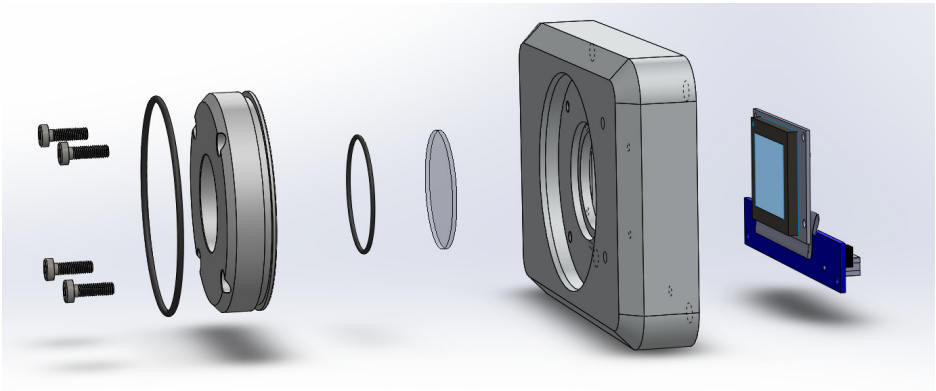


Figure 6.3.: Mechanical assembly of QT5022's head

6.4.2 *Body*

The Body is the mechanical group that holds all the other elements. Figure 6.4 shows a 3D model of the the body. On the figure, the elements have been colored with the same colors as Figure 6.1.

The green PCB is the Image Processing Pipeline, that interfaces the Processing Unit with a PCIe bus over a flexible PCB.

The blue PCBs conform the Processing Unit. It is composed by two different PCBs stacked one of the top of the other. The top one holds the power supply and connectors. The bottom one holds the memory, AMD Accelerated Processing Unit (APU), Ethernet adaptor and other chipsets. The bottom PCB can be used with other top PCBs to build cameras with a different form factor.

The red PCB is the adaptor for the Acceleration Unit. Any COTS Mezzanine card can be used.

The whole design is passively cooled through the case, which has a profile specially designed for dissipating heat. The main chips, i.e. FPGA, APU, memory and Ethernet adaptor are in direct contact to the case.

Once the body and the head are sealed, the camera can stand industrial environments with extreme conditions of heat, humidity and dirt.

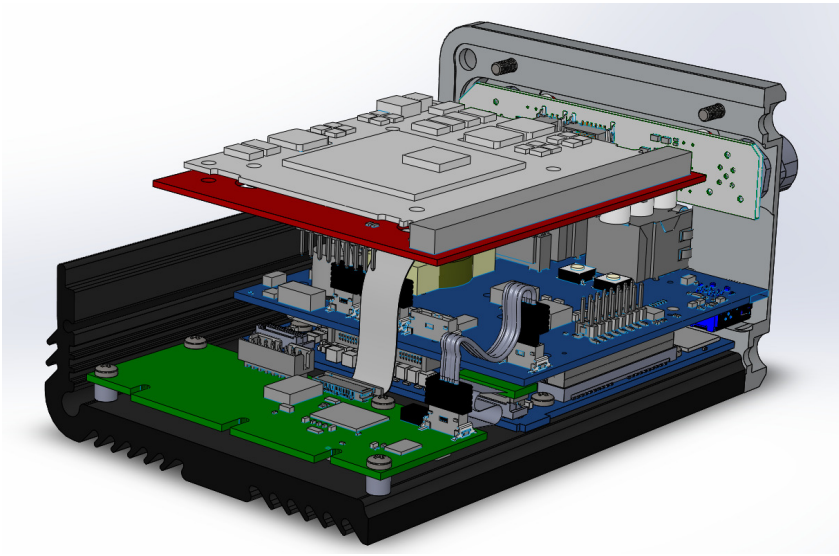


Figure 6.4.: 3D Model of QT5022's body

7

DATA FLOW

On the previous chapter, the proposed Computer Vision System has been described from a high level point of view. This chapters provides a behavioural description of the elements in the system.

The data transactions will be described from the sensor to the User Application. The reader will be introduced to the bare minimum Hardware and Software elements required to follow the discussion.

Contents

7.1 SENSOR

In a Computer Vision System, the sensor is dedicated to convert light into electrical signals. The outcome is produced in terms of pixels, that may comprise effective pixels, inactive pixels (i.e black reference) and broken pixels (dead pixels, lazy pixels).

From the nature of the electrical signals, imaging sensors can be classified in two groups: digital sensors and analog sensors.

Digital sensors produce discretized output that can be processed directly by the next element, in this case, an FPGA. Output comes in one or more channels.

Analog sensors, on the other hand, generate electric signals that require a digitization process before it can be processed by the FPGA. This discretization can be done by image specialised Analog to Digital Converters (ADs) or by general ADs. The former will take care of sensor timing and the blanking compensation.

Once the light has been converted into discretized processable data, it will have the form of one or more pixels streams and control signals, working at a typical speed of 10 to 100 MHz. E.g. the CMV12000, a CMOS chip [cmo15], which is compatible with the platform, can provide up to 64 pixel streams at 60 MHz and 10bits of bit resolution, i.e. 4.8 GiB/s.

7.2 IMAGE PROCESSING PIPELINE

The next element that the data encounters is the Image Processing Pipeline. On the QT5022, this is implemented on an FPGA.

An FPGA is an integrated circuit composed by millions of programmable logic units, denominated cells. A group of cells organized to make a function is denominated a core. Cores can be custom made or obtained from a third party. The different FPGA manufacturers offer a big selection of already-made cores with their design tools. On the following subsections, the different cores on the system will be described with special focus on the custom made cores. Figure 7.1 shows a dia-

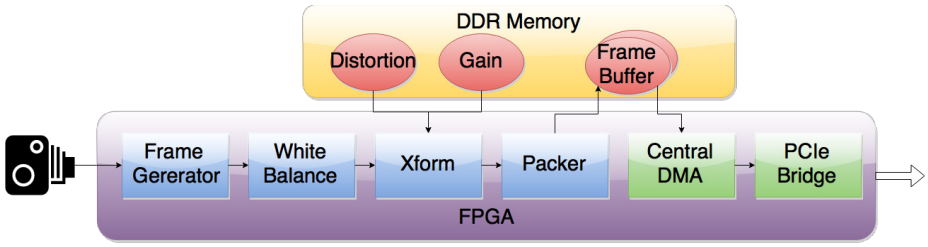


Figure 7.1.: Dataflow on the FPGA

gram of the dataflow on the FPGA. The elements on green are cores from the FPGA manufacturer, i.e. Xilinx, the elements on blue are custom made cores, and the elements on red represent memory locations on the external memory.

The architecture of an FPGA makes it very suitable for image processing [KL08].

7.2.1 *Frame Grabber*

The Frame Grabber is the core that fetches the pixel stream, reorders it, compensate the sensor noise, demosaic the image and does sensor dependent corrections like Fixed Pattern Noise.

This is the only core that needs to be designed specifically for each sensor type. The other cores, interfaces the data with a high performance custom bus, described later, that abstracts any sensors difference.

The Frame Grabber has been logically divided into Pixel Readout, FPN corrector, Binning and De-Mosaicer, described in the following subsections.

7.2.2 *Pixel Readout*

Modern sensors have resolutions that can reach tens of millions of pixels, with data depths of up to 16 bits per pixel and framerates of hundreds of

frames per seconds. This huge amount of data is transported in multiple high-speed interfaces like Low-voltage differential signaling.

Most of the FPGAs have specialized Input Output (IO) cells, that are capable of performing a high-speed readout. Unfortunately, those IO cells do not support natively the IO standard from all the sensors. To overcome this issue, custom synchronization algorithms needs to be implemented on the Frame Grabber core. A model synchronization algorithm is now detailed:

On the assumption of a sensor with multiple data channels, which are randomly delayed. The Frame Grabber needs to get rid of this delay, using a Delay Element and a Bit Slip.

The Delay Element can delay the data in steps as little as picoseconds for a finite number of steps. During calibration, each channel is configured to output continuously a train pattern, this train pattern must have a good variability of ones and zeros. The output from the sensor is evaluated for a period of time, if the data has been stable for the whole period, that step is considered stable. This is tested for every possible step of the delay element, generating a list of stable steps. A group of contiguous stable steps is called a window. The step at the center of the biggest window will provide the most stable data output. The calibrated step varies among channels and sensors.

The Bit Slip element can delay the data in steps of one bit. Once the Delay Element is calibrated, the Bit Slip will shift the data in until the output matches the train pattern.

When the calibration has been completed for every channel, the Frame Grabber can read the data from the sensor using the same readout clock.

Some sensors output the train pattern when they are not outputting valid data. It is a good practice to always verify the train pattern in these areas to compensate any temperature drift.

All the data from the different channels are saved into an intermediate buffer, and de-scrambled if needed by another process.

Figure 7.2.: Fixed Pattern Noise due to column amplifiers
The top image is the original image from the sensor, where vertical lines, product of the FPN, can be appreciated. The bottom image is the corrected image.

7.2.2.1 *FPN Corrector*

When light hits the sensors, the different pixels that compose the sensor provide an electric signal proportional to the amount of light. This signal needs to be amplified in the sensor, before it can be measured.

A common sensor architecture has one amplifier per column: due to the different linear characteristics of each amplifier, each column may have up to 1% of different response, but even if the differences are very small, their distribution makes it very perceptible for the human eye. An example of this effect can be seen in the Figure 7.2.

This effect, denominated FPN, can be fixed with linear transformation that can be described as:

$$P(x,y) = p(x,y) * g(x) + o(x) \tag{1}$$

where P(x,y) is the processed pixel at column x and line y, p(x,y) is the original pixel, g(x) is the gain for column(x) and o(x) is the offset for column x.

The FPN correction factor needs to be calibrated for every sensor. Even sensors from the same production batch will have different FPN characteristics. These properties are so unique that can be used to identify a specific sensor [LFG06].

7.2.2.2 *Binning*

The process by which two or more consecutive pixels of a sensors are summed together is called binning. The result image is smaller and lighter. Images taken with very small exposure time, or obtained with a very occlusive optic are good candidates for binning

Since the sum of the pixels may overflow, the sum is scaled by a value smaller than one. An horizontal binning of size two can be described as:

$$P(x,y) = S * (p(2 * x, y) + p(2 * x + 1, y)) \tag{2}$$

where P(x,y) is the processed image at column x and line y, S is the scaler and p(x,y) is the original image. P(x,y) is only defined for half of the original image.

Some sensors supports binning natively, in that case, the sensor implementation will always be preferred, because of the data reduction produced by the process [ZPF97].

7.2.2.3 *Demosaicer*

Humans perceive color through the cones in our retina. Cones are specialized cells that can sense light in different spectrum. A healthy eye contains three types of cones: Short, Medium and Long. Short cones sense blue, Medium green and Long red as seen of figure 7.3. Our brain will then process the information from these cells and we will perceive a particular color [Cuto6].

Sensors work as a wide spectrum cone. They give out a signal proportional to the amount of light that hits their surface.

In order to mimic the human perception, color sensors are covered with a color filter array. This array is formed by a mosaic of tiny filters that only let pass an specific range of wavelengths into a particular pixel.

There are different choices of filters arrays based on the purpose of the sensor, but the most common is the Bayer filter, named after Bryce Bayer, its inventor [Bay76]. The Bayer filter, as the human eye, contains three type of filters: red, green and blue. These three filters follow a 2x2

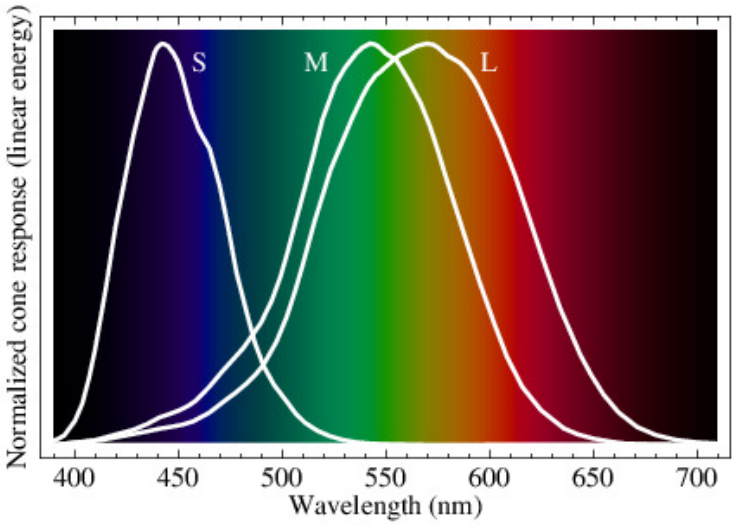


Figure 7.3.: Response of human cones

mosaic distribution. As an attempt to mimic our eye, there are two times as much green filters than red or blue filters.

Figure 7.4 from [jac15] shows an photo of a Bayer sensor taken with a microscope. The colored dots are the color filters.

Figure 7.5, shows the spectral response of monochromatic CMOS CMV-2000 sensor (black line), and the spectral response of three filters of the color version of the same chip. As it can be observed, the quantum efficiency of the color sensor is significantly worse than the monochromatic, and there are overlapping areas on the three filters.

The process by which the image from a mosaic camera is converted into a color image is called demosaicing. Demosaicing algorithms can be classified by the amount of result data.

- Decimation
- Interpolation

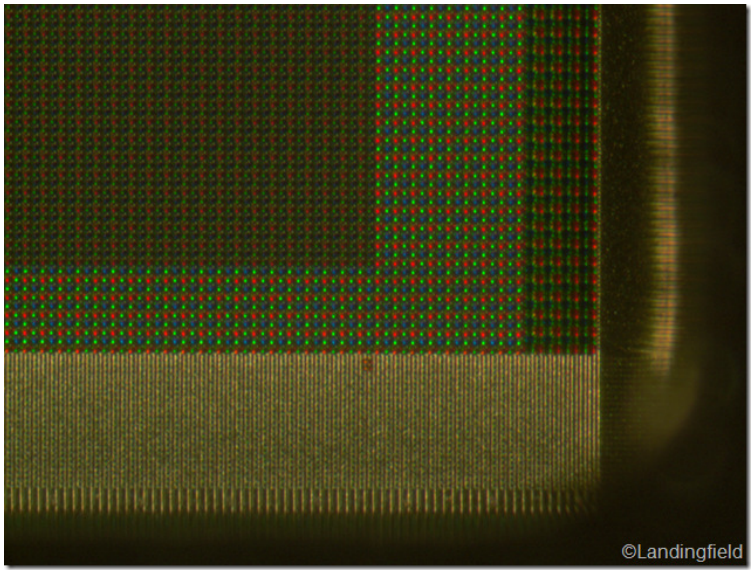


Figure 7.4.: Microscope image from a Bayer sensor

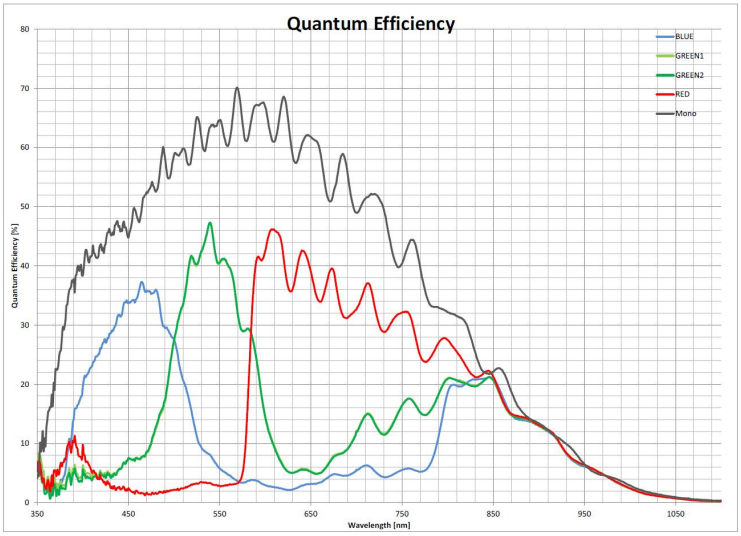


Figure 7.5.: Typical spectral response of a Bayer sensor

In decimation, every 2x2 RAW pixels block is converted into one color pixel. The formula for a normal 2x2 Bayer filter can be described as:

$$P(x,y) = [p(2x,2y), \frac{(p(2x+1,2y) + p(2x,2y+1))}{2}, p(x+1,y+1)] \quad (3)$$

where P(x,y) is the processed image at column x and line y and p(x,y) is the original image. P(x,y) is only defined in $1/4$ of the original image. In total, 25% of the data is lost and the process cannot be reverted without losses.

In interpolation, every RAW pixel generates 3 color pixels, their value is obtained by interpolating the surrounding pixels with the same color. Depending of the nature of the image, different interpolation algorithms can be used [LK01]. The result image is 3 times bigger than the original and some interpolation algorithms can be reverted.

Both decimation and interpolation produce artifacts on the final image, especially on the edges.

7.2.2.4 *Sensor merging*

Another technique to achieve a color image is to split the light into three different beams with a tri-chromatic prism. Every beam hits a different sensor which is covered by a color filter.

By multiplying the number of sensors, there is no more need to demosaic the image, and subsequently, there are no demosaicing artifacts.

Aside from the typical 3 color Red Green Blue (RGB) setup, prisms can be tailored to specific applications with more or less sensors. Due to production costs, it is rare to find configurations with more than 5 sensors.

The color image is obtained after fusing the monochromatic images from each sensor. This fusion is done at the optical and readout level. On the optical level, every sensor is calibrated so their difference is as small as possible. On the readout level, all the sensors needs to share the same pixel clock and control signals.

Sensor merging can be described as:

$$P(x,y) = [p^1(x,y), p^2(x,y)....p^n(x,y)] \tag{4}$$

where P(x,y) is the processed image at column x and line y, and $p^n(x,y)$ is the image from sensor n at column x and line y.

7.2.3 *Frame bus*

Once the data from the sensor has been read, the sensor-specific errors have been corrected and the color information has been decoded it is time to pass the frame to the rest of the cores that will do the sensor independent processing.

The interface between these cores has been standardized into a custom interface. The design criteria for this interface has been the following:

1. Simplicity: To ease up the creating of sensors, the interface should be as simple as possible, preferable based on well known standards.

2. Support for multiple channels: It should support not only RGB cameras, but also more exotic configurations with up to 8 different channels.

3. Reduced number of signals: The routing effort on the FPGA is proportional to the number of signals.

4. High color depth: Although commercial sensors usually work on 8 bit mode, industrial sensors can work in 10, 12, or even 14 bits mode. The extra depth plays an important role in the quality of the pattern matching algorithm.

As a result of this criteria this interface has been defined:

```
-- Channel number
FB_seq_out   : out std_logic_vector(2 downto 0);
-- Data
FB_data_out : out std_logic_vector(13 downto 0);
-- line data valid
FB_ldv_out   : out std_logic;
-- start of frame
FB_sof_out   : out std_logic;
-- end of frame
FB_eof_out   : out std_logic;
-- Pixel clock
FB_clk   : out std_logic;
-- Channel clock
```

Up to 8 different channels (colors) are serialized. seq_out is the index of the current color. data_out is the data in 14 bit mode, ldv_out tells when the bus contains data, and sof_out and eof_out are control signals, active at the start or end of frame respectively.

7.2.4 *White Balance*

Imaging sensors do not have a the same response for every wavelength, and the light might not be evenly distributed on sensitive regions of the sensor.

White balance is the adjustment of the different colors/channels present in an image to compensate the sensor spectral response and the scene lighting [CF06].

It can be modeled as:

$$P(x,y)[n] = p(x,y) * g^n(x) + o^n(x) \tag{5}$$

where P(x,y) is the processed pixel at column x and line y, p(x,y) is the original pixel, $g^n(x)$ is the gain for channel n and $o^n(x)$ is the offset for channel n.

On the proposed custom interface, thanks to the channel serialization, the same DSP block can be used for all the channels.

The properties of the light source might drift with temperature or aging. In order to have an accurate white balance, it can be controlled with a closed control loop: this is done with a gray reference plate on a known location of the scene.

7.2.5 *XForm*

The XForm is the main innovation of this Thesis in terms of imaging processing cores. It can be used for almost any light or geometric transformations, such as: even-out illumination, compensate lens distortion, rotate an image, fix perspective or re-sizing. All these transformations are done with just some milliseconds of latency. The simplicity of its design makes it susceptible for its synthesis in almost any FPGA. The XForm could be described as highly programmable remapper and bilinear interpolator.

The core input are: the image from the white balance and two user-defined parameter files: the distortion file and the gain file.

- The distortion file describes what group of four input neighbouring pixels is used for every output pixel, and how are combined.
- The gain file defines the output gain for every pixel.

7.2.5.1 *Parameter files*

The distortion file is an array of distortion structs. Its size is defined by the input image. The distortion struct is defined as:

```
struct distortion{
        __u16 col_res;
        __s16 col;
        __u16 line_res;
        __s16 line;
};
```

col and line are the column and line used for input data. col_res and line_res define the values for the interpolator. If col or line have negative values, the pixel will be skipped. All the lines need to have the exact amount of output pixels, and there should be at least one output line.

E.g. col=2, line=4, col_res=0x8000, line_res=0 means that the user wants the pixel (2.5, 4), i.e 50% of pixels (2,4) and (3,4), 0% of pixels (2,5) and (3,5).

The gain file is an array of unsigned bytes. Its size is defined by the output image, this is, the input size minus the skipped pixels. The value defines the gain for that particular pixel in 0.8 format. E.g a value of 0xff is equivalent to a gain of 0.996 and 0x80 is equivalent to a gain of 0.5.

These parameters files have a combined size of 9 bytes per pixel, i.e., an High Definition (HD) image (1280x720 pixels), would require more than 8 MiBs of space, which is allocated the external Random Access Memory (RAM) because it exceeds the Block Random Access Memory (BRAM) memory of the FPGA.

To reduce the burden of the data transfer, a compacted version of this parameter file is used. The compacted version is calculated automatically by the core driver. On the compacted version, every line is composed by a header data and one pixel data per column. The header is defined by the struct dist_header, that specifies the location of the first pixel of a line:

```
struct dist_header{
        uint16_t start_line;
        uint16_t start_col;
};
```

The pixel configuration is defined by the struct pixel data. cy and cx are the coefficients for the bilinear interpolator. Gain is the pixel gain, and move is the location of the next pixel using delta encoding, i.e. the distance to the previous pixel.

```
struct dist_pixel_data{
        uint8_t cy;
        uint8_t cx;
        uint8_t gain;
        uint8_t move;
};
```

This compacted format only requires 44.48% of the original size, i.e., 3.7 MiB of data for an HD image.

7.2.5.2 *Internal structure*

Because the parameter file are located outside the FPGA, they need to be pre-fetched. A process makes sure that a parameter Fist In First Out (FIFO) always have enough data.

The main part of the core is a dual port random access memory implemented on the FPGA BRAMs. Its size will define the resources requirement of the core and the latency of the core.

The left port of the BRAMs behaves as a FIFO. The input pixels are place in order into the memory; when there is no space left, the older data is replaced by the newest.

For every pixel written on the left port, four consecutive pixels are read from the right one. Which pixels are read is defined by the parameter file which has been previously decompressed.

These four pixels are passed to the bilinear interpolator, which fetches its parameters from the parameter decoder. The resulting pixel is multiplied by its specific gain, that is also decoded from the parameter file.

7.2.5.3 *Artifacts*

The following core constraints can produce artifacts on the output image:

1. Buffer size: At any given output pixel, the core buffer only contains a finite number of lines before and after that pixel. Accessing pixels beyond the buffer results in undefined behaviour.

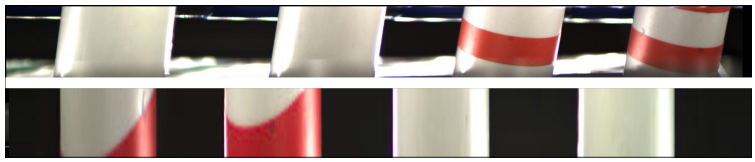


Figure 7.6.: Image processed with Xform
The top image is the original image from the sensor. The bottom image is the output of the xform core

2. Parameter format: To compact the parameter format, the position of every pixel is delta encoded. This means that its location is defined as the previous pixel plus an offset. In this case, valid offsets are [-4...3] if both directions. The only exception to this rule is the first pixel of every line, that is defined by an absolute value.

3. Blanking time: The random access pattern of the first pixel of every line requires a latency of 4 pixels. If the blanking are of the sensor is smaller than 4 pixels the output is undefined.

4. Output size: The core works in blocks of 4 pixels. If the output is not multiple of four, the missing pixels will have undefined output.

7.2.5.4 *Examples of use*

Figure 7.6 shows an example of use of the Xform core. The image was taken from a machine where the cameras, due to mechanical reasons, could be placed perpendicular to the targets and the illumination source were not at a constant distance to the four targets.

To find out the distortion parameters, a target with a chessboard pattern was placed on the lanes. The distortion map was calculated with the OpenCV camera calibration functions. The output from OpenCV was horizontally flipped, to get a mirrored output.

In order to obtain the light calibration, a grey target was placed in the lane. The inversion of the result image was used as gain parameter. The areas between the lanes were configured with a value of zero.

The core, just by using a gain map, and a combination of remapping and the bilinear interpolation, performs the following transformations with a latency of 3.2 milliseconds:

1. Rotation
2. Horizontal flip
3. Lens calibration
4. Perspective
5. Scale
6. Illumination

7.2.6 *Data Packer*

Images can be represented into multiple formats, defined by their colorspace, bit depth and channel order. The user application or the other hardware elements will define which is the right format in each case. The data packer is in charge of converting the data flow of pixels from the Frame Bus into the final pixel format.

Instead of limiting the amount of supported pixel formats to a small subset. the Data Packer follows a microprogrammed architecture that can support present and future pixel formats.

It is divided in four parts:

1. Colorspace converter
2. Data arranger
3. Endianness switcher
4. Memory interface

7.2.6.1 *Colorspace converter*

The data in the pixel bus is organized in channels. Each channel represent the intensity of a frequency band. In colorspace terms, this is referred as an additive colorspace format.

There are other colorspaces that take into consideration the human perception, like YUV, or formats for artistic photography, like grayscale or sepia.

The colorscace converter works as a transformation matrix that can linearly combine the input channels. The matrix is described as:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} * \begin{pmatrix} x^1 & y^1 & z^1 \\ x^2 & y^2 & z^2 \\ x^3 & y^3 & z^3 \end{pmatrix} + \begin{pmatrix} o^x \\ o^y \\ o^z \end{pmatrix} \tag{6}$$

Where X, Y and Z are the first, second and third channel of an pixel, x^2 is weight of the first channel for the creation of the second channel, and o^z is the offset for the third channel

Assuming that the first channel is the red, the second the green and the last the blue, the following values can be used to convert RGB into YUV according to ITU.BT-601[Rec11].

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} * \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & 0.368 & 0.071 \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \tag{7}$$

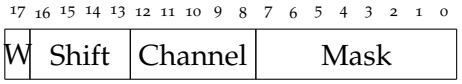
And these ones for converting from RGB into Grayscale according to Rec. 709 Luma function [Reco2].

$$\begin{pmatrix} Gray \\ Null \\ Null \end{pmatrix} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} * \begin{pmatrix} 0.21 & 0.72 & 0.07 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

7.2.6.2 *Data arranger*

Once that data is the right colorspace, it needs to be packed in the format required by the user. This is done by the Data arranger.

The Data arranger consist on a programmable Arithmetic Logic Unit (ALU) with an accumulator. The input is composed by 20 channels, and the size of the program is 20 instructions. Each instruction is 18 bit wide and is defined as follows:



where:

- W: Write the value from the accumulator to the output.
- Shift: Shift the value of the pixel to the left or to the right.
- Channel: Channel used (0-19).
- Mask: Nibbles to update on the accumulator.

In pseudo-code, each instruction can be described as:

```
Acc |= (Value[channel] >> Shift) & Mask;
if (W){
 *(Mem_out++) = Acc;
 Acc = 0;
}
```

E.g. The following code can be used produce a RGB24 image:

```
00) channel[00], shiftR  8, Mask 0x000000ff
01) channel[01], shiftR  0, Mask 0x0000ff00
02) channel[02], shiftR -8, Mask 0x00ff0000
03) channel[05], shiftR -16, Mask 0xff000000, DO_WRITE
04) channel[06], shiftR  8, Mask 0x000000ff
05) channel[07], shiftR  0, Mask 0x0000ff00
06) channel[10], shiftR -8, Mask 0x00ff0000
07) channel[11], shiftR -16, Mask 0xff000000, DO_WRITE
08) channel[12], shiftR  8, Mask 0x000000ff
09) channel[15], shiftR  0, Mask 0x0000ff00
```

```
10) channel[16], shiftR -8, Mask 0x00ff0000
11) channel[17], shiftR -16, Mask 0xff000000, DO_WRITE
```

7.2.6.3 *Endianness Switch*

Although the instruction set can produce data in both endianness, it requires twice as much as instructions to produce little endian data. E.g

Little Endian:

```
00) channel[00], shiftR  0, Mask 0x000000ff
01) channel[00], shiftR  8, Mask 0x0000ff00
```

Big Endian:

```
00) channel[00], shiftR  0, Mask 0x0000ffff
```

Due to the limited number of instructions, an Endianness Switcher has been implemented after the Data Arrager. If enabled, it can swap the output data bytes in groups of 16 bits. E.g: The output 0x12345678 will became 0x34127856.

7.2.6.4 *Memory interface*

The output data generated by the micro-instructions, is saved into a buffer implemented with BRAMs. This intermediate buffer cannot fit a whole frame, needing an external RAM.

The external memory has a latency in the order of hundreds of clocks, so in order to come up with the output bandwidth of the packer, it needs to be accessed in bursts. When there is enough data in the buffer for a RAM burst, a DMA process transfers the data to the external memory.

Once the whole frame is transferred to the external memory, an Interrupt (IRQ) is sent to the Processing Unit. Errors are also signaled via IRQ. E.g in the case of an overflow in the output buffer due to bus contention.

7.2.7 *DMA Engine*

The FPGA PCB has a 64 MiBs RAM memory, enough for holding 28 HD frames (1080x720 RGB24).

Although the RAM can be directly addressed by the CPU, the direct IO access presents a series of problems:

First of all, the data is only available for a certain amount of time. At a framerate of 40 frames per second, the frames are only valid for 700 milliseconds, before they are replaced by newer ones.

Then is the inefficiency of the Processing Unit accessing the FPGA. The Processing Unit can access its own memory orders of magnitude faster than the FPGA memory.

Finally, it is worth considering the OS memory model: the Peripheral Component Interconnect (PCI) memory is not part of the memory pool available for applications.

Therefore, in order to achieve the maximum memory performance, the frames needs to be moved to the CPU memory. This task is done by a DMA Engine.

In this case, the DMA Engine from Xilinx [Boo15] has been used. Xilinx Central DMA Controller (CDMA) supports Scatter Gather transfers with up to 256 user programmable descriptors and is configurable via a register interface. Each descriptor represent a memory transfer operation.

Descriptors could be located on any memory addressable by the core, but it is more efficient to locate them on a dedicated BRAM, this avoids memory contention. The maximum number of descriptors (256) fits in a 16 KiloByte 2^{10} bytes (KiB) BRAM block. Every descriptor has the following fields:

```
#define MAX_DESC_SIZE 8388607
struct cdma_desc{
        uint32_t next;
        uint32_t reservedo;
        uint32_t source;
        uint32_t reserved1;
        uint32_t dest;
        uint32_t reserved2;
        uint32_t length;
        uint32_t status;
```

```
        uint32_t reserved[8];
};
```

Where next is a pointer to the next descriptor, source is the source address, dest is the destination address, length is the transfer size (with a max size of 8388607 bytes), status is used as control/status register; and there are 11 reserved registers.

7.2.8 *PCI bridge*

The internal bus of the Processing Pipeline is Advanced eXtensible Interface (AXI) [Ste11], and the connection to the Processing Unit is done via PCIe. The core that bridges the two buses is Xilinx's AXI Bridge for PCIe [Boo45].

7.2.8.1 *Address Translation*

The address space of the Processing Pipeline and the Processing unit might have different sizes. In this implementation, the internal bus in the FPGA is 32 bits wide and the PCIe memory space is 64.

For PCIe to AXI, the core exposes up to 4 PCIe Base Address Registerss (BARs) to the Processing Unit. Every BAR is mapped 1 to 1 to an AXI region. In this implementation, there are two BARs: one is used for accessing the configuration registers of the cores and the other to map the FPGAs memory.

For AXI to PCIe, the core has a programmable address conversion table. Up to 6 AXI regions can be dynamically mapped into any location of the PCIe memory space.

This implementation has 4 AXI to PCIe regions, of 512 MiB each, giving a total size of 2GiB of addressable memory at any time.

7.2.8.2 *Bandwidth*

The PCIe core supports gen1 and gen2 PCIe and any number of PCIe lanes (i.e. 1x to 16x). The configuration of this implementations is gen1x1, which should provide up to a theoretical raw bandwidth of

2.5 GiB/sec which due to the 8b/10b encoding ends up delivering up to 250 MiB/sec. Unfortunately, the core only gave a throughput of 104/123 MiBs (FPGA to PC/PC to FPGA). After studding the transactions traces, it was concluded that the core was only producing data in 60% of the available slots. In collaboration with Xilinx, a bug was found in the core: due to a design error, all the memory transactions were serialized. This bug has been fixed on the version 1.03 of the core.

Once the bug was fixed, the result bandwidth was 188 MiBs (i.e. 75% of the theoretical maximum or 254 HD frames per second).

It is worth to notice that this speed is achieved using the maximum burst size in DMA. Programmed Input Ouput (PIO) access to this BARs has a transfer speed as low as a 300 KiB/sec. This is why in this implementation, frames are always moved via DMA, and PIO access is only used to access the register interface of the cores and uploading the Xform parameters files.

7.3 PROCESSING UNIT

In this implementation, the software abstraction used on the Processing Unit, to interface with the Image Processing Pipeline is Video4Linux2.

Video4linux2 is a very complex and rich framework capable of handling different set of devices, from USB webcams to PCI Television (TV) tuners. During the last years it has been re-designed to support composed devices like the media blocks present on some SOCs.

The use of Video4Linux2 for Industrial Applications is one of the most important contributions of this Thesis. Other systems [SH06b, JCP+10] in the literature have implemented their own custom acquisition libraries, the main problem with those libraries is the lack of standardization and the poor integration with other components of the system.

The author has collaborated with the community behind Video4Linux2, to extend its functionality towards a better support of Industrial Vision Applications. This work is detailed on the Chapter 9.

7.3.1 *Video4Linux2 Input/Output*

Video4linux2 provides the applications with 4 different methods for accessing frames:

- Read/Write: Data is provided sequentially by reading a char device. This is the simplest access scheme. Data is copied from the userspace to the kernel space with memcpy().
- MMap: The kernel allocates a number of buffers. Those buffers are shared with userland via mmap(). The ownership of the buffers and the signaling of when a buffer has valid data is done via IOCTLs.
- Userptr: Similar to mmap, but the buffers are allocated by the application via malloc(), or by other libraries like OpenCL.
- Dmabuf: Software abstraction for modeling the different DMA channels on the system.

Video device drivers implement one or more of those methods.

A helper library called Video Buffer 2 (VB2) [Cor11] is provided by the Video4Linux2 framework to abstract the video drivers from the different I/O methods. Thanks to this library, the video driver just implements one set of callbacks and VB2 will take care of the differences between IO methods.

The callbacks are detailed on the following lines:

```
struct vb2_ops {
int (*queue_setup)(struct vb2_queue *q,
    const struct v4l2_format *fmt,
        unsigned int *num_buffers,
        unsigned int *num_planes,
        unsigned int sizes[],void *alloc_ctxs[]);

void (*wait_prepare)(struct vb2_queue *q);
void (*wait_finish)(struct vb2_queue *q);

int (*buf_init)(struct vb2_buffer *vb);
int (*buf_prepare)(struct vb2_buffer *vb);
void (*buf_finish)(struct vb2_buffer *vb);
void (*buf_cleanup)(struct vb2_buffer *vb);

int (*start_streaming)(struct vb2_queue *q,
    unsigned int count);
void (*stop_streaming)(struct vb2_queue *q);

void (*buf_queue)(struct vb2_buffer *vb);
};
```

The most important callbacks are explained in a chronological order:

1. queue_setup: Called before the camera is started. It is used to negotiate the number of image buffers and their size. On this implementation it is checked that at least one frame can fit on the FPGA RAM.

2. buf_init: Called when the buffer is initialized, just after the memory is allocated by the framework. It is used to add metadata to the internal buffer structure. On this implementation, it is checked that the CDMA is capable of accessing the whole image buffer. At this point, CDMA descriptors are pre-calculated and saved as meta-data.

3. buf_queue: Called when the user wants to pass the ownership of a buffer to the driver. Once the buffer is queued, the user application cannot access it until it is returned to it. The driver will call vb2_buffer_done when the buffer is ready.

4. start_streaming: It is called by the user to start up the streaming. In this implementation, this callback starts the Image Processing Pipeline and the images are delivered at the required framerate to the FPGA RAM. Whenever there is a buffer available, the CDMA engine will copy the frame to the buffer and will notify the user. If there is no more space on the FPGA and there is no buffer available, the oldest frame will be discarded instead of stopping the pipeline. The user can find out this situation by inspecting the frame_number field on the buffer, or by checking the lost_frames control.

5. stop_streaming: Called when the frame flow is stopped. All the DMA transactions needs to be stopped and the buffers returned to the application. In this implementation, the Frame Buffer is stopped, the IOs are left tri-stated and the pixel clock is stopped. This is done to save as much power as possible.

7.3.2 *V4L2 Control Interface*

Video streaming is not only about buffer management and memory allocation, there are multiple settings on the camera that needs to be controlled. This goes from generic controls as brightness control, to more custom controls as the XForm controls.

Video4Linux2 provides an infrastructure for managing any kind of controls. This infrastructure, called v4l2-ctrl [Cor07], supports control aggregation, asynchronous event notifications, multidimensional controls, caching and custom typing.

On this implementation, all the custom functionality has been abstracted with v4l2-ctrl without requiring new IOCTLs or other custom

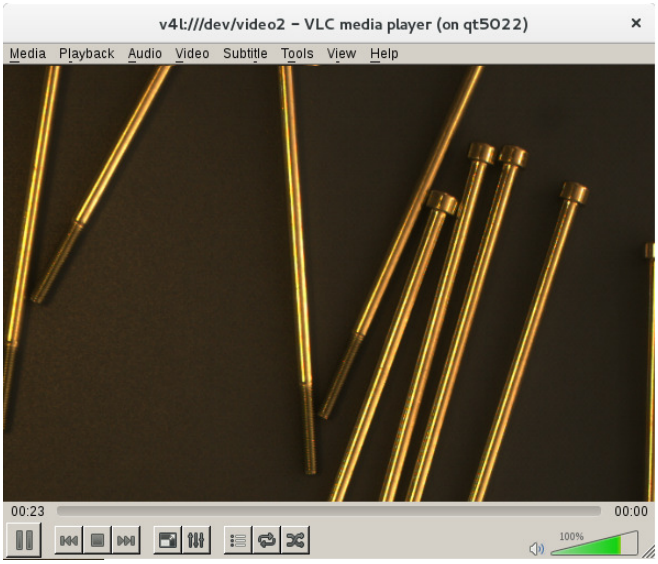


Figure 7.7.: VideoLanClient used to monitor a Screw counter

interfaces. As a result of this, any application or library that supports Video4Linux2 can be used to control the system.

On figure 7.7, VideoLan Media Player is used to display a screw counter.

7.3.3 *V4L2 Miscellanea*

V4l2 can also add time-code information to the frames. This metadata was originally used to display the time and frame information in the Video Cassete Recorders, i.e. the text showing the date on the corner of a frame. In this implementation, the time-code structure is used to export to userspace the mechanical position of the motors, i.e. encoder location. This information can be used to estimate the displacement of a frame and reduce the search size of some algorithms, e.g correlation.

Finally, Video4Linux2 provides a very accurate method for setting the framerate of a camera. This is used by this implementation to synchronize the capture with a conveyor belt or another camera.

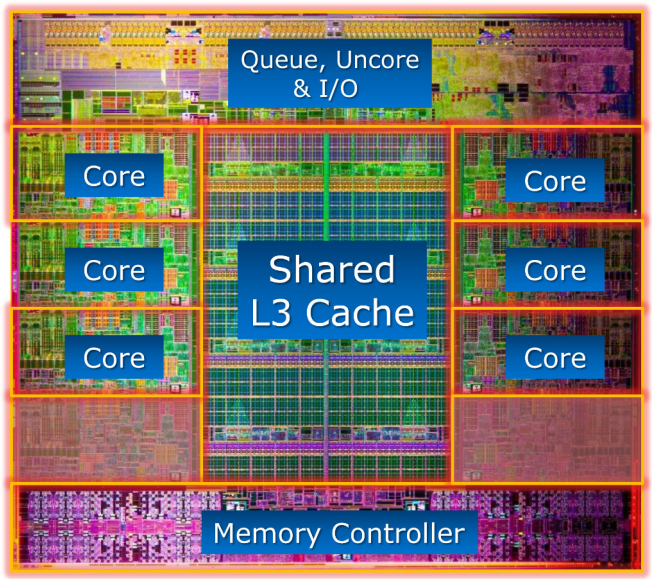


Figure 7.8.: Intel Core I7 Die

7.4 ACCELERATION UNIT

At this stage, the frame has been produced by the sensor, pre-processed, packed and moved into the Linux memory model by the Image Processing Unit.

On this implementation, the CPU on the Processing Unit can access the frames with two threads simultaneously, it has also a very complex cache for accessing the main memory that guarantees coherence among those two threads. The control flow of the applications is predicted with a jump unit in order to pipeline the instructions.

Figure 7.8 shows an the physical structure of a CPU, as seen on the figure, most of the silicon is used on tasks not directly related to computation.

The architecture of the CPU fits very well general problems, but has some disadvantages when processing images.

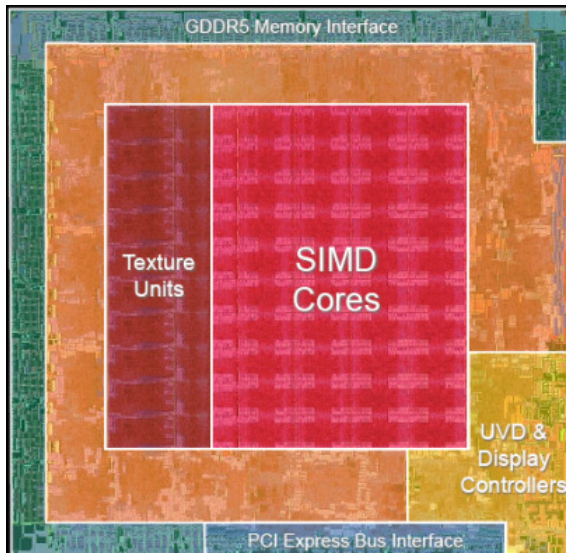


Figure 7.9.: AMD HD 4850 GPU Die

Images are composed by million of pixels that are usually processed in the same manner with a very predictable access pattern. This type of data processing can take more benefit from parallelization than from caching or jump predictions.

GPUs, like the AMD HD 4850 GPU (Figure 7.9), dedicate most of its resources in computing. As seen on the figure, most of the silicon is used for Single Instruction Multiple Data ALUs.

In the last years, these devices have been enabled for general applications and not only for output graphics [ND10].

The GPU do not have strict design constraints like the ones from X86 CPUs; but this this comes with a price: because GPUs do not follow the classical computing paradigm and they have to be programmed using specific languages as CUDA or OpenCL.

7.4.1 *OpenCL programming model*

In OpenCL [MGMG11] there are at least two devices, the host (i.e. the main CPU) and the target (i.e. the GPU).

The array of data is called NDRange. It can have multiple dimensions, and its maximum size is defined by the target.

The code that defines the operations over the NDRange is called kernel. It is the equivalent of an OpenGL shader.

The kernel is executed by multiple work groups. In a work group there is a number of work items. Every work item operate with an element of the NDRange and executes exactly the same instruction at the same moment.

7.4.2 *OpenCL memory model*

In OpenCL the memory model is divided in the following memories:

Private memory: it is only addressed by a work item. It would be the equivalent of a variable.

Local memory: it can be accessed by any work item of a work group.

Global memory: memory located on the GPU that can be accessed by any work group and the host.

Constant memory: same as the Global memory, but the work items cannot modify its content. It is the equivalent of the texture memory on OpenGL.

Host memory: Memory only accessible by the host (main CPU)

PCIe: memory located on the CPU that can be accessed by the GPU.

7.4.3 *Camera Data Flow*

Now that all the relevant elements of OpenCL have been described, the dataflow from Processing Unit to the Acceleration Unit, will be described.

1. The buffer memory is allocated with the OpenCL framework in an area addressable by the Accelerating Unit.

2. This memory is shared with Video4Linux2 via USERPTR interface.
3. The application enqueues the buffer.
4. Video4Linux2 fills the buffer with a frame and dequeues it.
5. The OpenCL framework copies the buffer into the Acceleration Unit.
6. The Acceleration Unit processes the frame.
7. The OpenCL framework copies the results back to the Processing Unit.

It is worth pointing out that on a video stream, these steps can be pipelined, reducing significantly the processing time.

8

HARDWARE

The previous chapter was focused on the different data transformations that happens on the system, from the sensor to the application.

This chapter describes the hardware used on the reference implementation of the Computer Vision System. Due to the modularity of the system, any of the components detailed on this chapter can be replaced with other module with the same interface.

Contents

8.1 EXTERNAL ACQUISITION ACCESSORIES

8.1.1 *Light Source*

Imaging sensors transform light into a measurable electric signal. This light might be reflected or emitted by the target. On most of the industrial applications, the target reflects the light from an artificial light source.

Some of the criteria for selecting the best light source for a particular application are: produced heat, emission spectrum, light intensity, consistency and working frequency [DSAG13].

- Produced heat: How much heat the light source adds to the system, this is a critical parameter because the noise level on the image sensor is proportional to the temperature.
- Emission spectrum: Each light source type have a characteristic emission spectrum, this is, the intensity of photons in each frequency. Figure 8.1 shows the spectra of different light sources.
- Light intensity: The luminous flux per area unit emitted by the sensor. High intensities allow the use of shorter exposure times on the sensor, reducing the motion blur [AR09].
- Consistency: How much are affected the properties of the light source by the aging or temperature. This is an important parameter for the acquisition consistency.
- Working frequency: Lights sources that are powered by mains electricity might be affected by its frequency, i.e. 50 Hz in Europe and 60 Hz. in USA. Light sources with low thermal capacity will flicker at the mains frequency.

This implementation has been tested with 5 different light sources: Natural Light, Incandescent, Halogen, Fluorescent and Light Emission Diodes (LEDs).

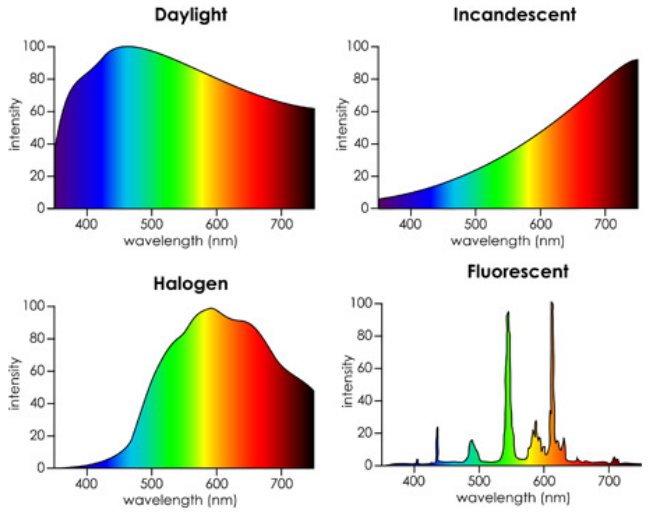


Figure 8.1.: Emission Spectrum of different light sources

8.1.1.1 *Natural Light*

The Sun is the most accessible light source. It does not use any power, covers almost all the spectra, delivers a huge amount of light and it is not affected by the mains working frequency.

Unfortunately, it is not very consistent, making it only a good option for open space applications like surveillance.

8.1.1.2 *Incandescent*

Incandescent light is produced by bulbs where a conductive metal that is heated up with an electric current until it glows. This has been the main illumination source for decades.

Although it has a very wide spectra it is not even, requiring a very aggressive white balance to get natural images.

At high sampling frequencies, the mains frequency is clearly visible, requiring advanced triggering or post-processing of the frames.

Although incandescent light produces a very powerful light, it is not very power efficient

As the light bulbs wore out, the emission spectra varies.

8.1.1.3 *Halogen*

Halogen light is produced by an incandescent filament surrounded by a halogen gas.

Halogen light covers a good part of the spectra with a lot intensity. Due to its high working temperature it is not very affected by the mains frequency. It also produces a very consistent light across its lifetime.

All this benefits, comes with the a cost of a high power consumption, i.e. heat.

Due to its wide spectrum and high intensity, it is the preferred light source on the initial steps of a project.

8.1.1.4 *Fluorescent*

A fluorescent light is produced by a low pressure mercury vapor gas lamp, that uses the fluorescence effect to produce visible light.

Fluorescent light provides a lot of light on some narrow bands of the spectra. The light properties are very consistent across the lifetime of the tube.

Like the incandescent bulbs, standard fluorescent tubes are very affected by the mains frequency. The industry has overcome this problem with the introduction of high speed tubes.

Fluorescent tubes produce much less heat than the equivalent incandescent or halogen light sources, but due to its very special emission spectra this light can only be used in specific applications.

8.1.1.5 *LED*

LED light is produced by a diode via the electroluminescent effect.

The frequency of photons emitted by the LED is determined by the band gap of the diode. Figure 8.2 shows the spectra of different LEDs, as seen on the figure, each LED has a very narrow spectra.

Multiple LEDs can be combined to get almost any spectra.

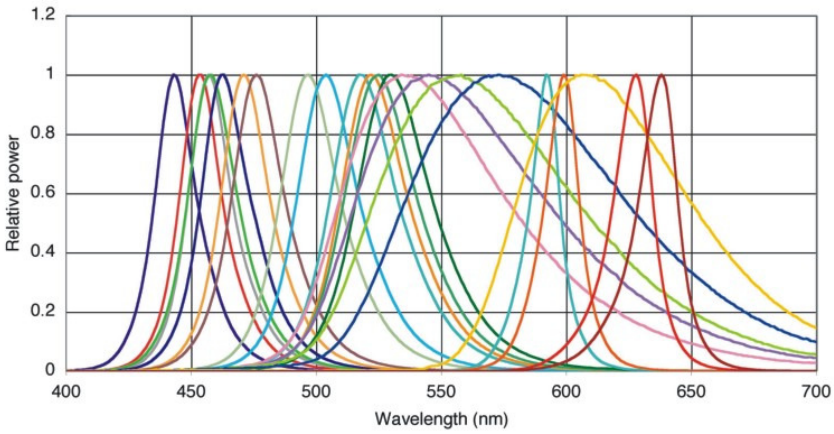


Figure 8.2.: Spectra of different LEDs

LEDs can be turned on extremely fast, allowing configurations where the light source in only enabled during the exposure time of the sensor, achieving very good power efficiencies.

This light source is not powered directly by the mains, and therefore it is not affected by its working frequency.

8.1.2 *Flash Unit*

This implementation has a custom external Flash Unit for LEDs. This unit, is in charge of post processing the flash signal from the camera and power the LEDs efficiently.

It is capable to control up to 8 different light sources with the input of up to 8 different cameras. Each light source is controlled by a small micro controller: the PIC16F1823 [p1612]. There is also a bigger PIC, the P18F87K22 [p1811], in charge of the communication with the Processing Unit.

This microcontroller family has been used because it is already present on other element of the system: the Hardware Monitor, detailed later on this chapter. The selection of the specific part was done based on the number of I/Os and ADs.

Figure 8.3.: Image from the The Frankencamera F2 with two Canon flash units attached

The flash signal is post processed following technical and legal limitations:

The heat sink mounted on the LED modules can only dissipate a fraction of the produced heat. This is not a problem if the LEDs are only enabled for a small fraction of time. The small PIC implements two different protections to avoid overheating. The first protection avoids too long flash pulses, and the second one accounts the total on-time over a long period of time.

The LEDs are high intensity blinking device that need to fulfill regulations against epileptic seizures [NSK+05]. To avoid the blinking perception, small pulses of light are emitted when the flash is OFF . For the human perception, the light seems constantly on.

The selected microcontroller has enough resources to be customized even further. It can output any desired flash pattern. This can be useful in experimental or creative photography. Other projects like the Stanford Frankencamera [ATP+10], have enabled support for this kind of flash in their systems. Figure 8.3 shows an image taken with the Frankencamera using an advanced flash pattern.

8.1.3 *Optics*

Optics are all the elements placed in front of the sensor that affect the light beam. Due to their sometimes hand-made manufacturing process, they can present more than 50% of the budget of a Computer Vision System.

8.1.3.1 *Lenses*

Lenses project the light beam into the sensor or into the next optic element [DSAG13].

In Industrial Computer Vision, most of the lenses used are fixed, this is, that their focal length cannot be adjusted. This is because they are more reliable against vibrations or extreme temperatures.

Variable lenses are usually relegated to the first stages of the applications. Some variable lenses contain a small motor that allows automatic focus or focal length.

This implementation has support for fixed and variable lenses with 3 different standards: C-Mount, Bayonete and MicroFourThirds.

MicroFourThird lenses can be controlled with a proprietary serial bus, without this bus, only manual lenses can be used; unfortunately its specification is not publicly available [KA12],

During the development of this thesis, Olympus have been contacted to get the specification of the MicroFourThirds protocol, which has been released under Non-disclosure agreement (NDA). This protocol has been implemented on an AVR microcontroller what serves as a bridge to the System Management BUS (SMBUS). Focal-Length, Aperture and Lens start/stop can be controlled by the user with Video4Linux2 controls.

8.1.3.2 *Spectrograph*

An spectrograph separates the frequencies forming a light beam. It takes a single dimension image and deliver a 2 dimensional image where every line (or column) is an specific frequency, as seen in figure 8.4

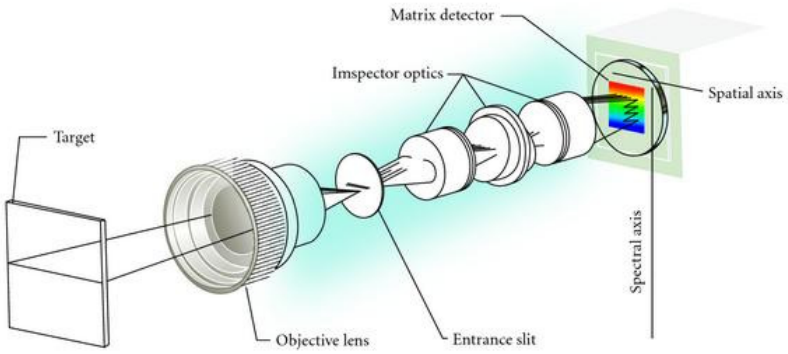


Figure 8.4.: Spectrograph, image courtesy of Specim

This implementation has been tested with a spectrograph from Specim (Finland) [Her15], which has been used in combination with an InGaAs sensor to create hyperspectral images.

The hyperspectral images, also referred as data cubes or cuboids, are obtained after stitching multiple images, just like a linear scanner stitches images to obtain a bidimensional image.

8.1.3.3 *Prism*

A dichronic prism is an optical device that splits the light in two different wavelength bands. Multiple dichronic prisms can be used to obtain more than two channels. Figure 8.5 shows a 3 Channel Prism.

Two custom prisms, manufactured by Optec, Italy, have been integrated with the platform.

The first prism is composed by three channels and can be used with C-Mount lenses. The second one has five channels and uses Bayonet Lenses. Both prism are used in combination with CCD sensors.

The manufacture of these prism is not an easy job: it requires expensive raw materials, good knowledge of the material properties and plenty of manual labour. These were the most relevant challenges encountered during the development of the prisms:

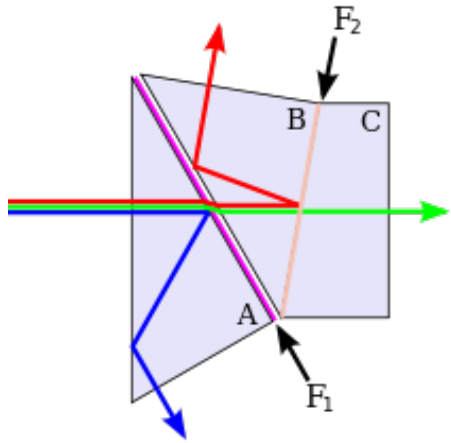


Figure 8.5.: 3 channel Prism

- All the channels needs to have the exact optical path, i.e. the distance between the entrance point of the prism and the sensor. If this is not achieved, each channel has a different focus.
- The sensors need to be placed manually on the prism, with a very small tolerance. An error bigger than 1/3 of a pixel of error, produces artifacts on the edges of the objects in the camera.
- Prisms are very fragile. A small impact can displace the sensors or even break the glass. Special procedures need to be followed on their transportation.
- Prisms need to be protected from external light with a black paint. If this is not done properly the external light can leak into the sensors.

8.2 IMAGE SENSOR

Image sensors convert light into a measurable electrical signal. They can classified by multiple criteria [DSAG13].

Based on the type of output:

- Digital: The output signal is already quantized and digitalized by the sensor. The output can be provided in serial or parallel interfaces. Fast serial interfaces, such as LVDS are commonly used on high pixel density sensor: Each LVDS pair can deliver up to 1.3 Gib/second.

- Analog: The output is an electrical signal, proportional to the light. It needs to be quantized by an external AD before it is processed by a digital system.

Based on the shutter type:

- Global Shutter: All the pixels are exposed to the light at the same time and for the same duration. Some global shutter devices supports readout while exposure, also known as pipelining.

- Rolling Shutter: All the pixels are exposed for the same amount of time, but at different times.

Based on the sensing technology:

- CCD: Spectral response goes from 200 to 1100 nanometers with a peak at 500 nanometers. CCD sensors provide a very good signal to noise ratio at the expense of higher power consumption.

- CMOS: Response is similar to the CCDs, but the peak is located at 700 nanometers. Signal to noise ratio is worse than CCDs, but consumes much less power and their price per pixel is better than CCD.

- InGaAs: Response goes from 800 to 1800 nanometers. Pixel density is significantly smaller than CCD and CMOS, with a typical resolution of 320x256 pixels. These sensors require special lenses, commercial lenses do not show a linear response on frequencies above 1100 nanometers.

- Amorphous Silicon: Response goes from 8000 to 14000 nanometers, i.e. thermal radiations are. They present a lot of defective

pixels due to its complex manufacturing process. These sensors are very expensive and have a lot of commercial restrictions due to their use in military applications.

Based on the filters they use:

- Monochrome: No color filter is placed in front of the camera. Commercial sensors usually mount a low-pass filters that blocks the infrared light and protect them with scratches. Sensors with absolutely no filter are sometimes marketed as Infrared-Enhanced.
- Bayer: A 2x2 mosaic filter is placed over the sensor.
- Multispectral: Have a complex pattern of filters disposed as a mosaic or as lines. They are made in small batches and have elevated prices.

The following sensors have been integrated with the platform:

8.2.1 *Sony ICX204*

Sony ICX204 [ICX] is a CCD Sensor with a resolution of 1024x768 pixels. It features an electronic global shutter with configurable exposure time. There are 43 extra columns and 9 extra lines of light insensitive pixels, i.e. black pixels, that can be used for real time noise compensation. The output from the pixels is presented in analog form, and the horizontal and vertical timing is controlled by external electronic. The pixel size is 4.65x4.65 micrometers, with a total chip size of 5.8 x 4.9 mm. Its maximum readout speed is 20MHz. There is a version with a Bayer filter.

A block diagram of the chip is shown in Figure 8.6

The vertical signals of the sensor are controlled by a Sony CXD1267AN [ICX] vertical clock driver. This chip is controlled directly by the Frame Generator on the Image Processing Pipeline.

An Analog Devices AD9970 [AD914b] 14-Bit CCD Signal Processor is used to control the horizontal timing and performing the AD conversion. This chip provides LVDS output and is configured via SPI. The AD9970

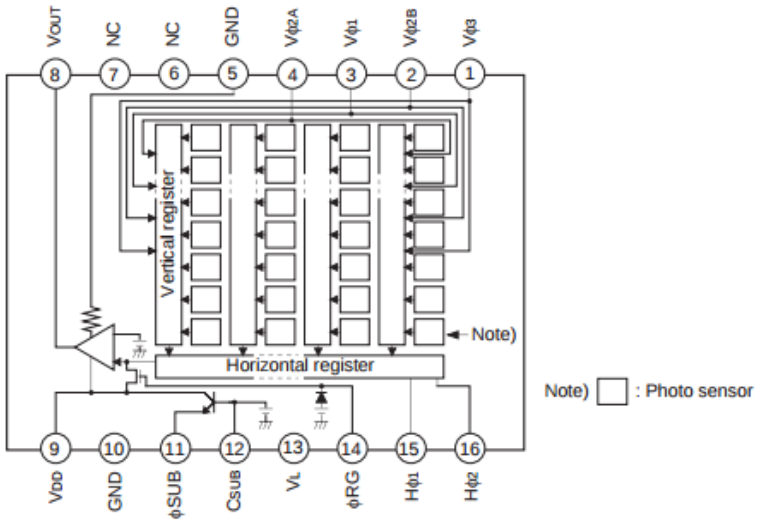


Figure 8.6.: Block diagram of Sony ICX204 CCD (from [ICX])

is a feature rich signal processor, with programmable gain and offset, black level compensation, programmable line skipping, Correlated Double Sampling (CDS) and a maximum conversion rate of 65 MHz.

This sensor has a framerate that can reach a maximum value of 23.56 Frames Per Second (FPS) at the maximum design readout speed. This speed can be overclocked to an out of spec value of 32.5 MHz in order to reach a framerate of 38.29 FPS. The framerate it is not dependent of the Window Of Interest (WOI), due to the internal structure of the sensor, every line needs to be read.

This implementation introduces a special readout scheme, that allows skipping 5 horizontal lines of pixels for every line. With this readout pattern a 640x480 image can take the same time as a 1077x542 image, i.e. 55.67 FPS instead of 38.29.

8.2.1.1 *Multiple CCD*

As mentioned before, there is a Bayer version of the CCD. Its use is limited to a small subset of applications, where its low resolution is not a handicap.

Applications that require higher resolution can combine multiple CCDs with a prism. On this configuration, each CCD is used for a specific color. This sensor is a very good candidate for this kind of configuration due to its small physical size.

The current implementation of the platform has support for one, three and five CCDs. The five CCDs version gives a very good performance on vegetable classifications, this is because the two extra channels are very sensitive to the chlorophyll.

8.2.2 *CMOSIS CMV*

CMOSIS CMV [cmo15] is a Family of CMOS sensors. It comes in different resolutions: from 0.0625 Megapixels to 70 Megapixels.

The current implementation of the platform supports the CMV2K (2048x1088), the CMV4K (2048x2048), the CMV8K (3360x2496) and the CMV12K (4096x3072). The supported sensors are available in mono, color (Bayer) and IR Enhanced versions.

These global shutter sensors can work at bit depths of 10 and 12 bits and their pixel size is 5.5x5.5 micrometers.

The CMOSIS sensors are configured via Serial Peripheral Interface (SPI) and produce a digital output in the form of multiple LVDS channels. The current implementation supports up to 16 LVDS data channels at a pixel clock of 30 MHz, that can deliver a maximum dataflow of 5.36 GiB per second. The synchronization to the data channels is done following the basic algorithm described on chapter 7, i.e. each channel is calibrated independently.

This sensor has a lot of embedded functionality, such as: pipelining, line skipping (reading 1 line of every N lines), multiple WOIs, simple decimation, programmable gain and offset, gamma correction, black sun correction, thermometer and horizontal and vertical flipping.

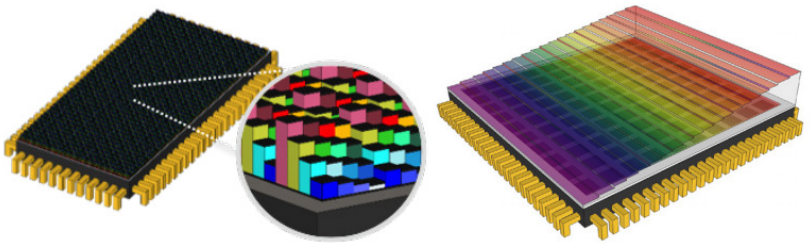


Figure 8.7.: Conceptual drawings of IMEC sensors

The CMV8K and the CMV12K also have black pixels that can be used for sensor calibration and online readout noise compensation.

The framerate of the CMV sensors is proportional to the number of lines on the WOIs.

The Belgium company IMEC [ime15] offers CMOSIS CMV2K sensors with advanced filters: The IMEC Hyperspectral mosaic imager is a 5x5 mosaic filter. The IMEC Scan Imager has 100 band-pass filters placed horizontally on the sensor. Figure 8.7, shows the conceptual drawing of the IMEC mosaic imager (left), and the line scan imager (right).

8.2.2.1 *Dual CMOSIS*

The current implementation of the platform supports a special head with two CMOSIS chips placed separately. This configuration has multiple applications:

Two identical standard optics can be combined to produce 3D-photography. This takes advantage of the separation of the two sensors, that mimics the human separation.

A hyperspectral line scanner can be combined with a monochromatic sensor. This is used to get a general view of a scene at the same time as the central line is analyzed in all the spectra.

A color sensor can be combined with a mono sensor with a pass-band filter, sensitive to the chlorophyll. This can provide similar results to the 5-CCD configuration at a fraction of the price.

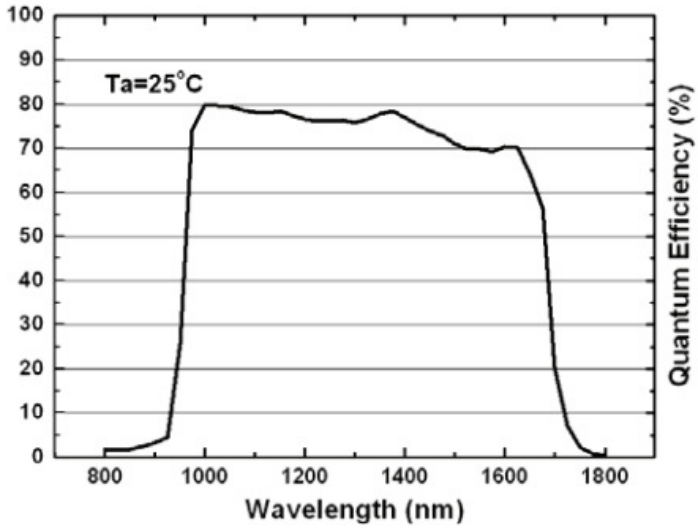


Figure 8.8.: Spectral response for Andanta FPA320x256-C

Lenses with different focal length can be mounted on each sensor. With this configuration, the central part of the scene can be analyzed in detail whithout losing the context information.

8.2.3 *Andanta FPA320x256-C*

Andata FPA320x256-C [fpa15] is an InGaAs sensor with 320x256 active pixels. Each pixel has a size of 30x30 micrometers. Its spectral response goes from 900 to 1700 nanometers as seen on figure 8.8. It has a pixel rate of 10MHz, proving a framerate of 122.07 FPS.

Like the CCD sensor, it provides an analog output, needing an extra AD for interfacing the LDVS input of the Image Processing Pipeline. This implementation makes use of the Analog Devices AD9259 AD, a 14 bit 315 MHz AD for this purpose.

The Andata chip requires a variable voltage reference, that in the current implementation is provided by the Analog Devices AD5628 Digital to Analog Converter (DAC) [AD514].

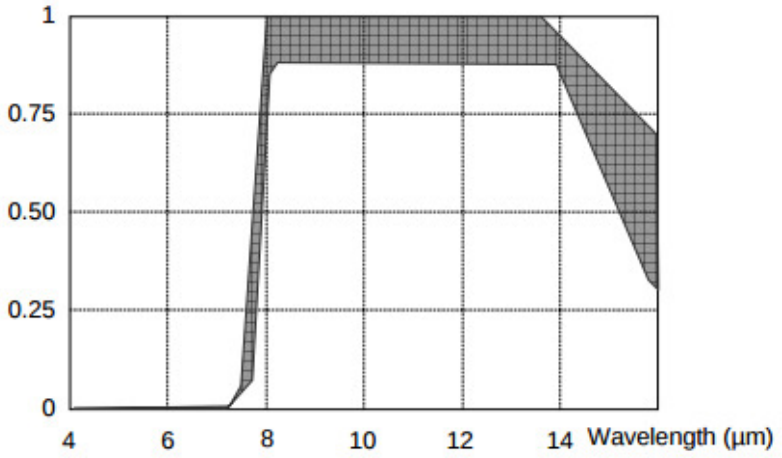


Figure 8.9.: Spectral response for Ulis UL 05 25 1-026

Every sensor is sold with a technical document describing its dead pixels, which are kept under a maximum of 1% of the total. On this implementation, this information is saved into a flash for automatic processing.

Due to its spectral response, this sensor is usually used in combination with a spectrograph to produce hyperspectral images.

8.2.4 *Ulis UL 05 25 1-026*

Ulis [uli14] UL 05 25 1-026 is an amorphous silicon microbolometer sensor. It has 1024x768 active pixels, with a pixel size of 17x17 micrometers. Its spectral response goes from 8000 to 15000 nanometers as seen of figure 8.9. It has an framerate of 50 fps and it does not need active cooling.

The Ulis chip has two output analog channels. On this implementation, the two channels are digitalized with an Analog Devices AD9259 AD [AD914a]. The same AD used for the Andanta InGaAs sensor.

Each sensor is delivered with its own quality report contract, enumerating the dead pixels on the chip this report is delivered under a very restrictive NDA.

This sensor is mainly used on thermal applications, due to its spectral response.

8.3 IMAGE PROCESSING PIPELINE

On this implementation, the Image Processing Pipeline is developed with an FPGA. Table 3 summarises the properties of the FPGA versus other alternatives for the Image Processing Pipeline.

Table 3.: Comparison of different devices used as glue logic

	FPGA	Micro-controller	ASIC
Number of pins	900	100	As required
LVDS support	YES	NO	YES
Time resolution	nanoseconds	microseconds	nanoseconds
Multiple voltage	YES	NO	YES
Time to market	Weeks	Days	Years
Prototype cost	200$	20$	100.000$
Re-Engineering	bitstream	firmware	New design
PCIe interface	YES	NO	YES
Performance/Watt	Good	Bad	Excellent

An FPGA is a device built around an array of configurable elements, named cells, that can be configured to behave as any digital circuit. The configuration of the cells is kept in a non volatile memory, read at boot time by the FPGA. The basic cell is composed by a Look Up Tables (LUTs), but there are also specialised cells in charge of IO, DSP, memory or clock generation [MC07].

Digital acquisition systems can take a great advantage of the versatility of the FPGAs when they are used as glue logic. FPGAs can interact with sensor with time restrictive interfaces, with accuracy down to nanoseconds.

FPGAs have hundreds or even thousands of IO pins, arranged in groups with different IO standards. The IO pins of the FPGA support

LVDS natively, which is the proposed interface of this platform between the Sensor and the Image Processing Pipeline.

FPGAs can change their behaviour with a change of their configuration. This is an excellent property when they interface state of the art and experimental sensors that constantly change their documentation.

In terms of power, FPGAs, consume much less power than an equivalent processor, this is particularly useful with sensors that are very sensitive to temperature changes like the CMOSIS sensors. In these sensors, the dark current will double with every $6 - 7°C$ increase: In a sensor sunning at $51°C$, the dark current will be 2000 e/s, but at $5.5°C$, it will only be around 15.6 e/s.

The cells on an FPGA can be reconfigured at runtime, this reconfiguration can happen with different levels of granularity.

The DSP cells presents on the FPGA can be used to implement multiple image processing algorithms "on the fly".

Finally, the development cycle of an FPGA is much shorter than the development cycle of a custom ASIC. This reduces the integration time of a sensor in a platform and the time to market of the device.

This implementation uses a Xilinx Spartan6 [spa15] LX100T. This FPGA family presents the best price/performance ratio and is optimised for high-speed IO. In terms of resources, is the second biggest FPGA of its family, with over 100.000 logic cells, 180 DSPs and 976 KiB of distributed RAM.

The usage report of the FPGA shows that the selected FPGA has still plenty of resources left for implementing more cores:

```
Device Utilization Summary:

Slice Logic Utilization:
  Number of Slice Registers:              22,730 out of 126,576   17%
    Number used as Flip Flops:            22,664
    Number used as Latches:                    4
    Number used as Latch-thrus:                0
    Number used as AND/OR logics:             62
```

```
  Number of Slice LUTs:                    24,501 out of  63,288   38%
    Number used as logic:                  21,851 out of  63,288   34%
      Number using O6 output only:         15,621
      Number using O5 output only:            443
      Number using O5 and O6:               5,787
      Number used as ROM:                       0
    Number used as Memory:                  1,950 out of  15,616   12%
      Number used as Dual Port RAM:         1,072
        Number using O6 output only:          492
        Number using O5 output only:           30
        Number using O5 and O6:               550
      Number used as Single Port RAM:           0
      Number used as Shift Register:          878
        Number using O6 output only:          639
        Number using O5 output only:            1
        Number using O5 and O6:               238
    Number used exclusively as route-thrus:   700
      Number with same-slice register load:   558
      Number with same-slice carry load:      141
      Number with other load:                   1

Slice Logic Distribution:
  Number of occupied Slices:                9,496 out of  15,822   60%
  Number of MUXCYs used:                    4,768 out of  31,644   15%
  Number of LUT Flip Flop pairs used:      29,460
    Number with an unused Flip Flop:        9,163 out of  29,460   31%
    Number with an unused LUT:              4,959 out of  29,460   16%
    Number of fully used LUT-FF pairs:     15,338 out of  29,460   52%
    Number of slice register sites lost
      to control set restrictions:              0 out of 126,576    0%


IO Utilization:
  Number of bonded IOBs:                      110 out of     296   37%

Specific Feature Utilization:
  Number of RAMB16BWERs:                      194 out of     268   72%
  Number of RAMB8BWERs:                        19 out of     536    3%
  Number of ILOGIC2/ISERDES2s:                 22 out of     506    4%
```

```
Number of IODELAY2/IODRP2/IODRP2_MCBs:      24 out of     506    4%
Number of OLOGIC2/OSERDES2s:                59 out of     506   11%
Number of BUFPLLs:                           1 out of       8   12%
Number of BUFPLL_MCBs:                       1 out of       4   25%
Number of DSP48A1s:                         13 out of     180    7%
Number of ICAPs:                             1 out of       1  100%
Number of PCIE_A1s:                          1 out of       1  100%
Number of PLL_ADVs:                          4 out of       6   66%
Number of STARTUPs:                          1 out of       1  100%
```

On the current implementation, there is a different bitstream for each sensor configuration plus a sensor agnostic bitstream, loaded when the system starts. As soon as the OS has probed the head, the FPGA is reconfigured with the specific sensor bitstream. This coarse grain reconfiguration model [provides the best utilization of the resources on the FPGA.

A second Image Processing Unit, based on a Xilinx Kintex FPGA [kin15] has been integrated in the system. This FPGA is supported by the new generation of Xilinx Tools: Vivado. One of the new features introduced by Vivado is the High Level Synthesis, that allow programming of the FPGA with C or C++. The productivity with this languages is much higher than with Hardware Descriptor Languages such as VHDL or Verilog [SWC04].

At this moment, all the heads are supported by the Spartan board, and the Kintex board only supports the CMOSIS sensor. On the near future, both boards will support the same range of sensors.

8.4 PROCESSING UNIT

In this implementation the Processing Unit is implemented with an APU. Table 4 summarises the criteria for this selection, comparing this solution with a dual chip X86 solution (Nvidia discrete GPU and Intel CPU) and an ARM SOC, a popular combination of a GPU and an ARM CPU mainly targeted for mobile phones [AMY09].

An APU is a combination of CPU, North-Bridge chipset and GPU on the same chip. It is specially targeted for embedded devices and

Table 4.: Comparison of different computation units

	AMD APU	Dual Chip	ARM SOC
Tools availability	+++	+++	+
Low Power Consumption	++	+	+++
PCB Simplicity	++	+	+++
Computation Power	++	+++	+
OpenCL Support	+++	+++	+

In each field, more + is better

notebooks. The last generation consoles (Xbox One and Playstation 4) are driven by APUs [Mag, Hyn].

The APU chosen has been the G-T65N [Inc13] from AMD, with 2 bobcat microarchitecture CPU cores at 1.65GHz, a GPU RADEON 6320 at 500 MHz and an AMD Northbrige with with memory, PCIe and Serial AT Attachment (SATA) interfaces. Despite its great capabilities, its Thermal Design Power is as little as 18W [Inc13].

8.4.1 *CPU*

The selected APU has a dual core 1.65GHz 64 bit X86 architecture CPU. The X86 architecture is used on most of the consumer computers. Thanks to its great availability, there is a great availability of debugging, development and tracing tools. Some of these tools will be described on chapter 8, but at this point it is worth to mention the tools that were used during the development of the hardware:

- Coreboot [Boro9]: an open implementation of a Basic Input/Output System (BIOS). It has been particularly useful for testing the memory and power management parameters.

- Memtest86+ [Bra]: an open memory tester. This tool has been executed on the platform for weeks to validate the reliability of the main memory of the system.

- AMD validation tools. These tools, obtained under NDA, perform platform specific tests to measure the performance of the platform under thermal stress.

8.4.2 *GPU*

The selected APU has a GPU RADEON 6320 at 500 MHz.

This GPU shares its memory with the CPU, which reduces the number of memory copies during the data process. This results in very low latency image processing applications.

One of the benefits of this GPU is its great support of the OpenCL language [TNI+10]. AMD has been one of the early adopters of this technology.

8.5 HARDWARE MONITOR

The selected APU requires a external device for powering up the system and monitoring its supplies. On this system this has been implemented with a programmable microcontroller.

The microcontroller used is a MicroChip PIC18F26K22 [p1811], It has 64 KiBs of Flash, 4 KiB of RAM, 2 Master Synchronous ports that can be used to communicate via Inter-Integrated Circuit (I2C) or SPI, ADs, support for self-programming, Programmable Brown-out Reset, 2 Universal Asynchronous Receiver/Transmitters (UARTs) and Timers. Figure 8.10 shows the interface of the device with the rest of the platform.

The programming capabilities of the microcontroller are used to provide the following advanced functionalities:

- System Power Up: The device makes sure that the power source is stable, and the clocks are present before it proceeds to power up the peripherals and the other elements in order.
- Watchdog: The main CPU must communicate periodically with the Hardware Monitor, otherwise the whole platform is restarted. This can be enabled or disabled on demand via SMBUS.

- Voltage Watchdog: The input voltage is constantly monitored for power glitches or under voltage. The whole system is restarted in order on the event of a voltage error.

- Voltage monitor: The different voltage values can be readout via I2C with the main CPU.

This extra functionality could not be obtained with a COTS hardware monitor like the LM Texas Instruments [lm9].

8.6 PERIPHERALS AND INTERFACES

The implemented Processing Unit has the following peripherals and interfaces:

- One Mini PCIe port: This interface is a combination of 1xPCIe, USB and SMBUS on the same connector. It is usually available on notebooks and there is a wide selection of device that support this interface, such as 4G modems, WIFI cards, USB interfaces.

- Four serial ports: The current implementation system has mounted a Fintek F81216A [fin] chipset with support for RS232 and RS485.

- One SATA port: The connector is located under the case and can be used to connect a optical drive or a hard drive.

- One CFast port: This interface is a variation of Compact Flash with integrated SATA connectivity instead of IDE/PATA.

- 2 Gigabit Ethernet: The current platform has a Intel 82580 chip with 2 interfaces, each one capable of delivering up to 133 MiB/sec. The chip has hardware support for Precise Time Protocol (PTP).

- 2 USB ports: One of the ports supports the EHCI Debug standard, that can be used for low level debugging of the BIOS and the Linux Kernel.

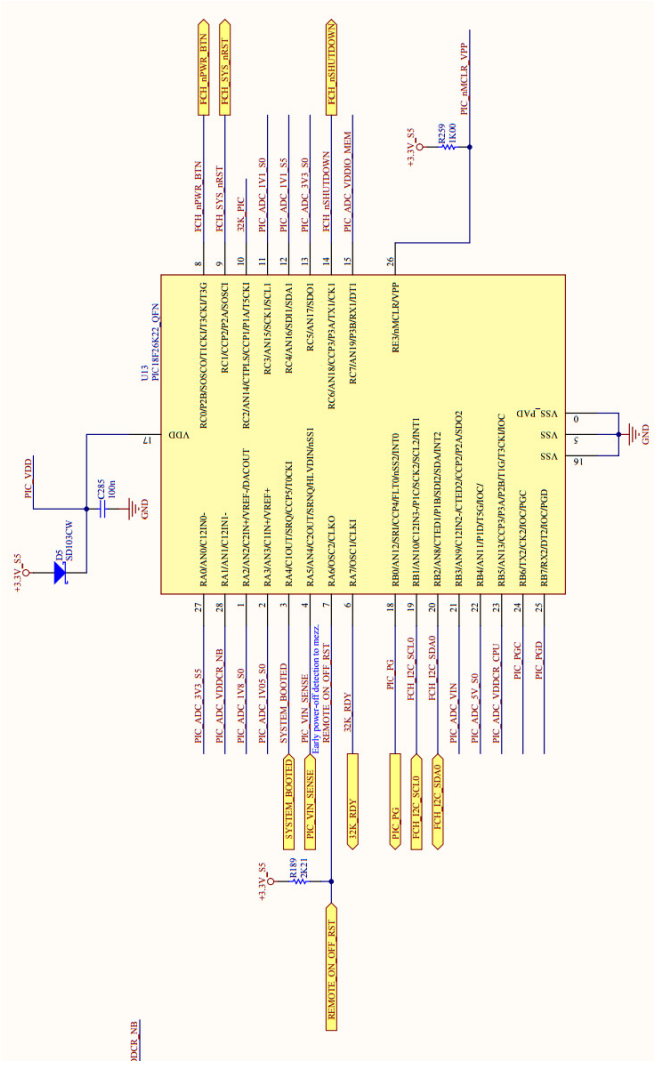


Figure 8.10.: Hardware Monitor Schematic

- 6 LEDs: They are used to have a visual indication of the status of the device. Three of the LEDs have fixed functionality: Power connected, ON status and SATA; the other 3 are user programmable.
- 6 IOs: They can be used as Flash output or Trigger input. The signal delay from these IOs to the sensor is less than 6 nanoseconds.

8.7 ACCELERATION UNIT

The current implementation supports an Acceleration Unit for compute demanding applications. The Acceleration Unit is connected to the platform with a PCIe Mezzanine connector.

Acceleration Units tested on the system have been AMD GPUs, because of its great great OpenCL support.

When using the Acceleration Unit it is worth noticing that the extra computation power comes with a thermal cost. The extra heat will affect the imaging sensor signal to noise ratio and the performance of the CPU: that will step down its frequencies at high temperatures.

The proprietary driver from AMD has an integrated power governor that partially powers off the GPU when it is not used. Unfortunately, AMD's driver does not provide any performance counters to evaluate the power consumption / heat dissipation. Without this information it is impossible to create power-efficiency guidelines for the platform.

An experiment has been set up to obtain this information:

The hardware of the experiment consists on a Tektronix A622 100 Khz Current probe that measures the main power supply of the camera. The probe is connected to a Tektronix MSO7104A High End Scope. This scope can integrate the current measurements over a programmable period of time. Figure 8.11 shows the output of the scope.

The software of the the experiment is conformed by a modified version of the Matrix Multiplication reference design from AMD [Sta11]. In this version, a programmable delay is added between the runs. using this application, the following loads have been characterized:

1. 0% Load: A camera in idle state.

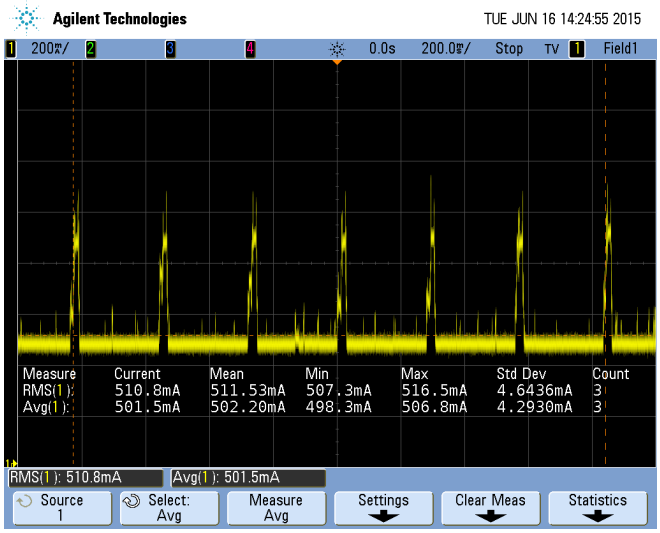


Figure 8.11.: 9% Load 275 msec period

2. 100% Load: big Matrix (142 mec)
3. 50% Load big Matrix: 142 msec on / 142 msec off
4. 9% Load big Matrix: 142 msec on / 1420 msec off
5. 100% Load: small Matrix (25 mec)
6. 50% Load small Matrix: 25 msec on / 25 msec off
7. 9% Load small Matrix: 25 msec on / 250 msec off

The performance is measured in Giga FLoating-point Operations Per Seconds (GFLOPs) only when the GPU is active. The results are shown in the Figure 8.12.

The relative power consumption. Follows the following formulae:

$$Power_{GPU} = MEAN(Current_{measured}) - MEAN(Current_{idle}) \quad (9)$$

where MEAN() is the mean function over 2 seconds. $Current_{measured}$ is the instant value of the current while running the experiment and

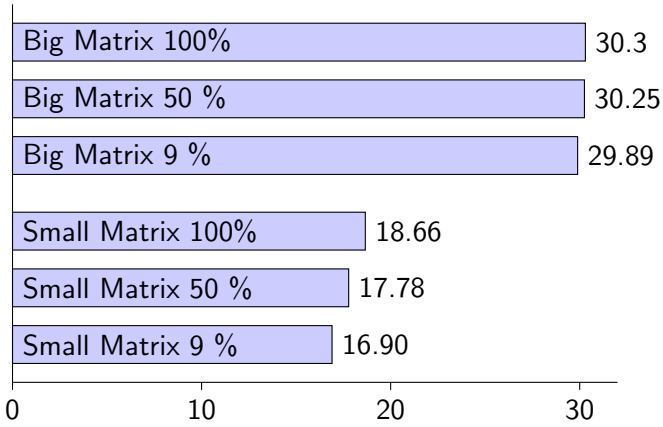


Figure 8.12.: Performance in Gflops

$Current_{idle}$ is the instant value of the current when the system on idle state.

$$Power_{Efective} = Power_{GPU} / Load \tag{10}$$

Where load, is the load in parts per unit.

$$Power_{Relative} = Power_{Efective} / MEAN(Current_{max} - MEAN(Current_{mean})) \tag{11}$$

Where $MEAN(Current_{max})$ is the mean of the current at max load over two seconds.

$Power_{Relative} > 1$ Will mean that the GPU wastes power on the stepping. An ideal governor will provide a 100% performance with a $Power_{Relative}$ of 1.

The $Power_{Relative}$ for the different experiments can be seen in Fig. 8.13

In order to obtain the performance per power, the measured performance is divided by the relative power. Obtaining the results shown in Fig. 8.14.

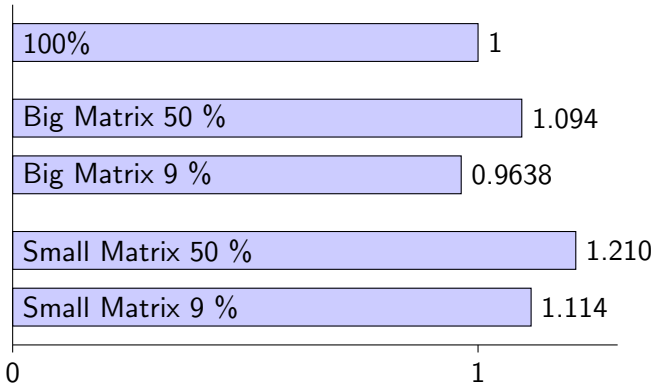


Figure 8.13.: Relative power in parts per unit

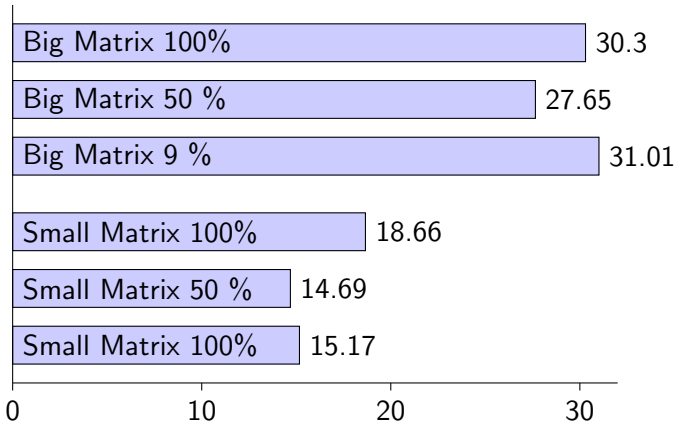


Figure 8.14.: Performance per power in Gflop

8.8 INTERACTION WITH OTHER SYSTEMS

8.8.1 *PTP*

The current implementation provides Trigger and Flash IOs for its easily synchronization with external systems.

The applications that require a more advanced trigger sequence relay on a common clock, which is synchronized via Ethernet.

Legacy Network Time Protocol (NTP) [Mil91], can provide accuracy down to milliseconds, but that is not enough for image acquisition. To understand why NTP can not deliver a better performance, it is worth analyzing how it works:

1. Device A gets the time from its internal clock (to)
2. Device A sends a package P with to to Device B
3. Device B receives the package
4. Device B gets the time from its internal clock (t1)
5. Device B process the package
6. Device B gets the time from its internal clock (t2)
7. Device B sends the package P with to,t1 and t2 back to device A
8. Device A receives the package
9. Device A gets the time from its internal clock (t3)

The round trip delay of the system is: $(t_3 - t0) - (t_1 - t_0)$
The offset is $\frac{(t_1 - t_0) - (t_2 - t_3)}{2}$

In this protocol, the accuracy of the offset and the round trip is affected by all the jitter between the time is obtained from the internal clock and the packet is delivered (1 and 2, 3 and 4, 6 and 7, 8 and 9). This jitter is produced by the context switching and the Ethernet buffer processing.

The industry has proposed a new time synchronization protocol,PTP [LEWM05]. PTP shares the concepts of round trip delay and offset

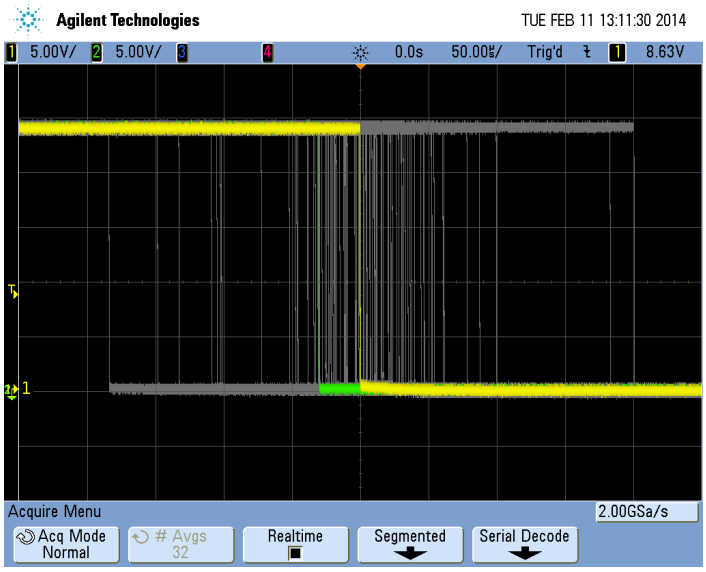


Figure 8.15.: NTP Synchronization

with PTP but uses specialized hardware to timestamp the packages as it leaves or arrives the network interface. This way the calculations are immune to the jitter.

In order to validate the accuracy of PTP and NTP in this implementation, the following experiment has been designed:

Two cameras have been used, the flash signal of each camera is connected to a high end scope (Tektronix MSO7104A). One of the flash signals is used as trigger for the scope, which is configured with infinite persistence.

Each camera has an application with real time priority that triggers a frame at the beginning of every second.

On the first run, both cameras are synchronized with NTP. Figure 8.15, shows the results of the experiment. The jitter from the first camera to the second goes from -175 to +200 microseconds.

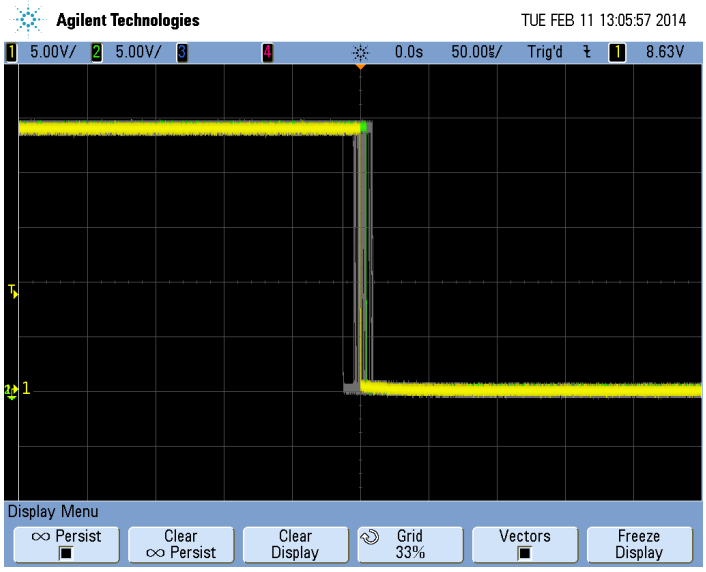


Figure 8.16.: PTP Synchronization

On the second experiment, cameras are synchronized with PTP (Figure 8.16). In this case, the delay from the first camera to second goes from -10 to +10 microseconds.

8.9 ETHERCAT

Industrial Computer Vision Systems are usually used in conjunction with other sensors and actuators. A good example of this is the Celox Potato Grader from Newtec (Fig. 8.17) [New], this machine features 6 conveyor belts, 512 GPIOs for the drop system (fingers), 12 vibrators, 12 pneumatic flow controls, 12 infrared proximity sensors and 12 encoders.

Although all the devices on the industrial application could be wired with normal cables this is an impractical solution due to the high number of devices and the electrical noise of an industrial environment.

Ethernet, on the other hand, is a better option in terms of noise isolation and less installed wires. Unfortunately standard Ethernet proto-

Figure 8.17.: Potato Grader

cols like Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) do not have the required properties in terms of latency and jitter: The latency is induced by every device that needs to interact with a package, as it needs to read the whole package before it can send it to the next device. The jitter is created on the device FIFOs.

Industrial Ethernet protocols are designed to overcome these problems. One of these protocols is Ethernet for Control Automation Technology (EtherCAT) [JB04]. In EtherCAT, the sensors and actuators are connected with a ring topology and are capable of processing the Ethernet packages on the fly. Solving the jitter and latency problems of TCP and UDP protocols.

The current implementation of the platform fulfills all the hardware and software requirements of EtherCAT. The following EtherCAT devices have been tested on the system: Encoder, Serial Port, Digital Input and Digital Output.

9

SOFTWARE

On the previous chapters, the hardware (chapter 8) and the dataflow (chapter 7) have been described. In this chapter, the different elements of the software stack will be detailed following a bottom-up approach, i.e. from the elements that are closer to the hardware to the more abstract layers.

The initial goal was to use Open Source whenever possible, as they would provide the system a standard API, but most of the standard packages for image acquisition were not designed for Industrial Applications.

Instead of generating new software elements, the author has collaborated with the Open Source community to extend the scope of their tools. As a result of this work, a great amount of code have been shared with the community, and accepted into their codebase after peer-review.

Contents

9.1 KERNEL

In the Operating System field, kernel is called to the piece of software that abstracts the hardware to one or more applications. A modern kernel is capable of providing at least the following services: memory management, devices management, implementation of system calls, provide multiuser and multitask abstraction, and implement mechanisms for ensuring fair and safe resource sharing among applications and users.

The kernel used in the proposed Computer Vision System is Linux, an Open Source kernel that follows the design principles of Unix and is widely used in embedded devices, supercomputers, desktops and consumer electronics, such as phones or televisions. According to the kernel patch statistics [kps15] Linux is being developed by more than 14000 persons, with an average of 34 contributions each.

Table 5 compares Linux with other kernels. The main reason for selecting Linux for this platform has been the availability of its Source Code, its wide hardware support and the great community of developers behind it.

Table 5.: Comparison of operating system kernels

	Linux	Free-BSD	Windows
License	Open-Source	Open-Source	Proprietary
Supp. Architectures	9	31	1
Code Availability	YES	YES	NO
Support	Excellent	Good	Scarce
Price	Free	Free	$\approx$ 90$/license
Debug and Profiling	Excellent	Good	Medium
No. Applications	Excellent	Good	Good

As direct result of this research, more than 170 patches have been accepted into the upstream version of the Kernel. This places the author of this Thesis as the second most active contributor from Spain and one of the top 5% globally [kps15]. These patches are referenced on the appendix of this Thesis.

9.1.1 *Linux Development Model*

Linux is an Open Source project where all the development is done publicly and is peer reviewed [kpa15].

When a developer finds a bug or wants to add new features, divides the changes into logical steps, denominated patches. The patches are formatted into "Unified Diff" format [AKM+08] and the developer signs the patch approving their inclusion into an Open Source Project according to the Linux Kernel Certificate of Origin [cog15].

The patches are then shared with the community through thematic mailing lists. Anyone on the list can test them, propose changes, acknowledge or not-acknowledge each patch. The author of the patch is expected to reply to all the comments with a justification or a new version of the patch.

The Kernel is divided into areas, denominated subsystems, with one responsible per area, denominated Maintainer. Once there is consensus about a patch, the Maintainer signs the patch and adds it to his copy of the Kernel, denominated tree.

All the modifications from all the Maintainers are then included into a common tree, denominated linux-next, where modifications are tested and benchmarked with hardware from different manufactures.

Once the modifications have been tested, the main responsible from the Linux Kernel, Linus Torvalds, fetches the changes from linux-next into his own tree, signing the commits on the process.

Not every type of patch is accepted at any time. At the beginning of each release cycle, Linus Torvalds accepts improvements and bug-fixes into his tree, this is called a Merge Window [KCM08]. After some time, only bug fixes are accepted, closing the Merge Window.

As a result of this elaborated Development Model, a standard modification requires up to 8 months of peer review and testing before is released to the public.

The Kernel Summit conference is held every year to discuss structural changes on the Kernel and the development model. Access to this conference is only granted after a nomination process. Some of the

Video4Linux2 contributions from this Thesis were discussed in the Multimedia workshop of the Kernel Summit 2013 held in Edinburgh.

9.1.2 *Linux Device Driver Model*

The Linux Kernel specifies a driver model [CRKH05, Moc02]. This model is used to generically represent and operate every device in the system. The Linux Device Driver Model is formed by different elements:

- Bus: represent hardware buses like PCI or USB.
- Device: each element of the bus like a sound card or a mouse.
- Class: description of a functionality implemented by a device in a specific topic. E.g: led class, GPIO class or sound class.
- Module: software element that implements one or more classes for a set of devices. Eg. The Intel Gigabit Ethernet driver.

The user can interact with these abstractions through a virtual filesystem called sysfs [Moc05].

The devices are created in different ways. On enumerable buses, like PCI, the devices are probed at boot time, on hot-plug buses, like USB, the devices can be generated and destroyed at any time. For the buses that cannot be enumerated, the kernel must be provided with some kind of device list. The most common ways to describe a non-enumerable bus are :

- platform devices: Static array that is passed to the kernel. A higher level module list all the devices that is aware of. E.g. the X86 core driver instantiates the standard serial ports that are accessible through ioports.
- user enumeration: The user may instantiate devices through the sysfs filesystem. There is a very limited devices-type that can be user instantiated. An example of this are the hardware monitors located at the SMBUS.

- device tree: A device tree is a Binary Large OBject (BLOB) describing a group of devices, their resources and their relation [LB08]. This BLOB can be located in a file or in memory. E.g: ARM devices make extensive use of the device tree to describe the different SOCs.

Platform devices can be implemented very easily, but each combination of hardware requires a specific module or even a specific kernel. Device trees, on the other hand, need much more effort to be implemented, but the same modules can be used on multiple hardware combinations.

The Linux Kernel natively supports Device Trees on the PowerPC and ARM architecture. On both architectures the Device Tree is implemented as a single immutable device tree. A new API is being introduced to support runtime changes in that tree [dyn15].

9.1.2.1 *Linux Device Driver Model and QT5022*

On the QT5022, the data acquisition and pre-processing occurs on an FPGA that is connected to an X86 CPU via PCIe. That FPGA is reconfigured based on the connected sensor.

The natural way of abstracting this structure, following the Device Driver Model consists on a single driver taking care of all the functionality of the FPGA. This abstraction presents some problems:

- A different model of FPGA would require a completely new module, even if they share great amount of cores.
- The kernel module needs to support all the possible combination of sensors.
- All the functionality of all the cores is loaded in memory even if it not going to be used.
- The code is less modular and more difficult to test and debug.

In order to overcome all these problems a novel approach based on Device Trees has been taken. This solution is based on a bridge driver, small changes on the driver subsystem and one driver for every core.

9.1.2.2 *The Bridge Driver*

PCIe devices are identified by a Vendor and a Product ID that are accessible through the PCIe configuration space [BAS04].

On the kernel, each module has a list of supported IDs, this list is evaluated every time a new device is discovered. If the device matches a module ID-list, the module takes ownership of the device.

In this Implementation a Bridge Driver has been implemented with support of both the Spartan and Kintex FPGA. When the FPGA is started, it loads a sensor-agnostic bitstream from the primary partition of its Flash. This bitstream is not capable of communicating with the sensor, but has functionality for accessing the FPGA Flash, the head Electrically Erasable Programmable Read-Only Memory (EEPROM) and the PCIe bus.

As soon as the driver takes ownership of the device, it reads the EEPROM and the appropriate bitstream is located on the filesystem and compared with the one installed on the second partition of the FPGA; if they differ, the new bitstream is installed on the flash. The FPGA is then reconfigured with the bitstream on the second partition.

While the FPGA is being reconfigured, the PCIe link is down, forcing the downgrade of the bus speed. As soon as the FPGA is reconfigured, the PCIe link needs to be retrained to achieve its maximum speed.

Now that the FPGA is configured with the sensor specific driver, the Bridge driver reads from the filesystem a device tree BLOB with a description of the different cores in the FPGA. The Bridge Driver will add this device tree to the main device tree and the new devices will be probed by their specific drivers.

This excerpt of the kernel log shows the process described on this section:

```
[ 6.976204] qt5023 0000:01:00.0: of: Using dtb firmware file:
qtec/qt5023/AXI_LX100T_BASIC.dtb
[ 6.979693] qt5023 0000:01:00.0: intc: allocated at 0xb0000000
with IRQ 36
[ 6.998227] m25p80 spi32766.0: Quad already enabled
[ 6.998241] m25p80 spi32766.0: s25sl064p (8192 Kbytes)
```

```
[ 7.006931] 4 ofpart partitions found on MTD device spi32766.0
[ 7.006944] Creating 4 MTD partitions on "spi32766.0":
[ 7.006954] 0x000000000000-0x000000800000 : "Full FLASH"
[ 7.007237] 0x000000000000-0x000000380000 : "Golden Bitstream"
[ 7.007414] 0x000000380000-0x000000700000 : "User Bitstream"
[ 7.007627] 0x000000700000-0x000000800000 : "Production Data"
[ 7.007835] xilinx_spi b0010000.spi-flash: at 0xB0010000
[ 7.503282] qtec_pcie b0030000.pcie_bridge: Axi-pcie bridge supervisor
[ 8.512066] qt5023 0000:01:00.0: Outdated mtd2,
Upgrading with qtec/qt5023/AXI_LX100T_CMOSIF-MONO.bin
[ 8.512079] qt5023 0000:01:00.0: DO NOT POWER OFF!!!!!!!!!!
[ 8.512084] qt5023 0000:01:00.0: Upgrading header of mtd2....
[ 8.512088] qt5023 0000:01:00.0: Erasing....
[ 9.822664] qt5023 0000:01:00.0: Writing....
[12.441171] qt5023 0000:01:00.0: Upgrading body of mtd2....
[12.441185] qt5023 0000:01:00.0: Erasing....
[44.333691] qt5023 0000:01:00.0: Done with mtd2
[45.339857] qt5023 0000:01:00.0: of: Using dtb firmware file:
qtec/qt5023/AXI_LX100T_CMOSIF-MONO.dtb
[45.374958] qt5023 0000:01:00.0: QT5023 camera head (build id #25) ready!
[45.914469] qtec_cmosis-0: 16 chans: Found a 11 units length eye
with center in 14 [4444000004444444]
[45.914658] qtec_cmosis-0: qtec_cmosis fg core
version 0x18 (CMV2000v2 Mono) V4L2 subdevice registered as qtec_cmosis-0
[45.914962] qt5023_video b0060000.packer_0: Using apertures 0-3
[45.915424] qt5023_video0: qt5023_video V4L2 device registered as video0
[45.916941] qtec_xform-0: qtec_xform version 0x22 size 147456 pixels
[45.917650] qtec_white-0: qtec_white V4L2 subdevice
[45.918704] b0070000.uart_0: ttyUL0 at MMIO 0xb0070000
(irq = 47, base_baud = 0) is a uartlite
[45.918882] qt5023_video0: Unable to probe m43 controller.
Continuing without it
[45.967653] qtec_mem qtec_mem: qtec_mem: Allocated 0x4000000 bytes
[45.985641] qt5023_video0: 3 formats supported
```

9.1.2.3 *Changes on the driver subsystem*

Although most of the device tree functionality already existed on the kernel it could not be built for X86-64 platforms. Additionally, accessing

the device tree functionality in ways that were not initially designed for, revealed bugs in some untested code paths.

This Thesis has contributed to the port of device trees to X86, and have also fixed several bugs on the device tree core modules.

9.1.2.4 *Core Specific Drivers*

Due to the bus abstraction provided by the Bridge Driver, each core can be handled by a different module. The Image Processing Pipeline is composed by two types of cores: standard cores by Xilinx like the PCIe bride or the DMA engine; and custom cores like the Frame Generator.

Some of the Xilinx cores were already supported by the Linux Kernel but their drivers were designed to be accessible only by the embedded processors on the FPGA (PowerPC or ARM). As a result of this Thesis, the modules for the SPI core, the GPIO core and the UartLite were modified to support the X86 platform. These changes were contributed to the community and merged with the upstream version.

The modules for the custom video processing cores will be detailed on the following sections.

9.1.3 *Video4Linux2*

As introduced on the chapter 7, the Kernel provides a video input/output framework. This framework, named Video4Linux2, provides a clean and robust user-space API [DVR99].

This system, in contrast to the devices described in the chapters 4, 3 or 5, makes use of Video4Linux2. The use of an standard API provides compatibility with third party libraries and guarantees the portability of the applications.

Imaging sensors have become a key element in consumer electronic devices: from mobile phones, to tablets and notebooks. The markets pushes every day for smaller products with high pixel density at a competitive price. The manufacturers tries to cope with the market requirements replacing expensive optics and imagining sensors with post-processing of the image. Post-processing is done in the SOC in

different blocks forming complex pipelines [Ban13]. Video4Linux2 has evolved through the years to support these complex devices with the media framework.

Although the Image Processing Pipeline dataflow, described on the chapter 7, follows quite well the design model of consumer electronic devices, it has some singularities that could not be directly mapped into Video4Linux2 elements.

During this thesis, there has been a great effort to extend the Video4Linux2 capabilities to support the requirements of this thesis. This work started with a proposal on the mailing list, that due to its scope required a physical discussion on the Multimedia Summit of the Kernel Summit 2013 and later in the Linux Plumbers 2014. After public review, the key points of the proposal have been accepted and merged into upstream.

Now, the key parts of the implementation using Video4Linux2 will be described.

9.1.3.1 *Main Module*

The Main Module, is in charge of implementing the Video Device Node that will be used by the user to fetch images and configure the camera. It is build upon the Video4Linux2 helper libraries: VB2 and v2l-control.

Before fetching images, the user must negotiate with the Main Module the format and framerate of the data acquisition. Figure 9.1 shows the same scene captured with three different pixels formats. This format must be agreed by all the cores in the dataflow. Video4Linux2 describes the pixel formats with Four Character Code (FOURCC) identifiers [Wil]. Thanks to the Matrix Packer core, described on the chapter 7, the following FOURCCs are supported:

- V4L2_PIX_FMT_RGB24: 24 bits RGB format, 8 bits per channel.
- V4L2_PIX_FMT_RGB32: 32 bits RGB format, 8 bits per channel plus 8 padding bits.
- V4L2_PIX_FMT_BGR24: V4L2_PIX_FMT_RGB24 with a different channel order.

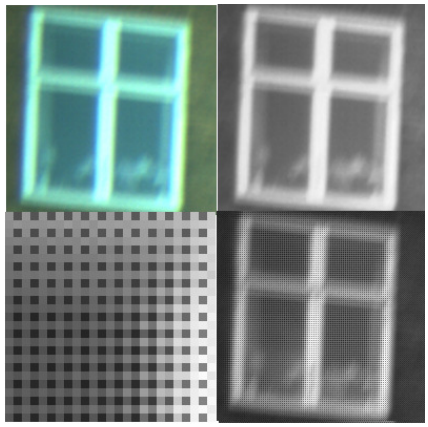


Figure 9.1.: Different pixel formats. Top left: RGB24; Top right: GREY; Bottom right: SBGGR8; Bottom left: Detail of the SBGGR8 image

- V4L2_PIX_FMT_BGR32: V4L2_PIX_FMT_RGB32 with a different channel order.
- V4L2_PIX_FMT_GREY: 8 bit grayscale format.
- V4L2_PIX_FMT_Y16: 16 bit grayscale format, arranged in Little Endian.
- V4L2_PIX_FMT_Y16_BE: 16 bit grayscale format, arranged in Big Endian.
- V4L2_PIX_FMT_SBGGR8: 8 bit format of an unprocessed Bayer Sensor starting on the Blue pixel.
- V4L2_PIX_FMT_SGBRG8: Similar to SBGGR8, but the Bayer mosaic starts on the Green-Blue pixel.
- V4L2_PIX_FMT_SGRBG8: Similar to SBGGR8, but the Bayer mosaic starts on the Green-Red pixel.
- V4L2_PIX_FMT_SRGGB8: Similar to SBGGR8, but the Bayer mosaic starts on the Red pixel.

- V4L2_PIX_FMT_UYVY: 16 bit UYVY 4:2:2 format. Useful for MPEG encoding.
- V4L2_PIX_FMT_YUYV: 16 bit YUYV 4:2:2 format. Useful for MPEG encoding.
- V4L2_PIX_FMT_QTEC_RGBPP40: 40 bit, 5 channel format, 8 bits per channel.
- V4L2_PIX_FMT_QTEC_RGBPP80: 80 bit, 5 channel format, 16 bits per channel.

The format V4L2_PIX_FMT_Y16_BE was not originally included in Video4Linux2. As an original contribution of this Thesis, the format was added to Video4Linux2. This format allows the creation of netpbm 16 bit images [MV96] with no further processing.

The formats V4L2_PIX_FMT_QTEC_RGBPP40 and V4L2_PIX_FMT_QTEC-_RGBPP80, represent the raw output of the Frame Bus, and they are too specific for being interesting to the community.

All the other formats are part of Video4Linux2.

Supporting a great amount of formats is crucial to guarantee compatibility with third-party libraries and limit the number of format conversions in the pipeline.

Once the format is negotiated with the user, it is time to initialise the buffers that will contain the images. Depending on the dataflow, buffers will be initialized by the user or by the kernel. Initially, VB2 didn't support user allocated buffers for scatter gather devices. As a result of this Thesis, VB2 has been extended to support that type of allocation. This work has been included into Video4Linux2.

Both the DMA Engine and the PCIe bridge have some constraints regarding the locations of the buffers in the memory. As a result of this Thesis, the allocation logic on VB2 was modified to prefer compacted memory, which is more likely to fulfill the DMA constraints, this modification has been accepted into the Kernel.

The Video Processing Pipeline has a high degree of customization by the user, with hundreds of controls. Figure 9.2 shows the list of available

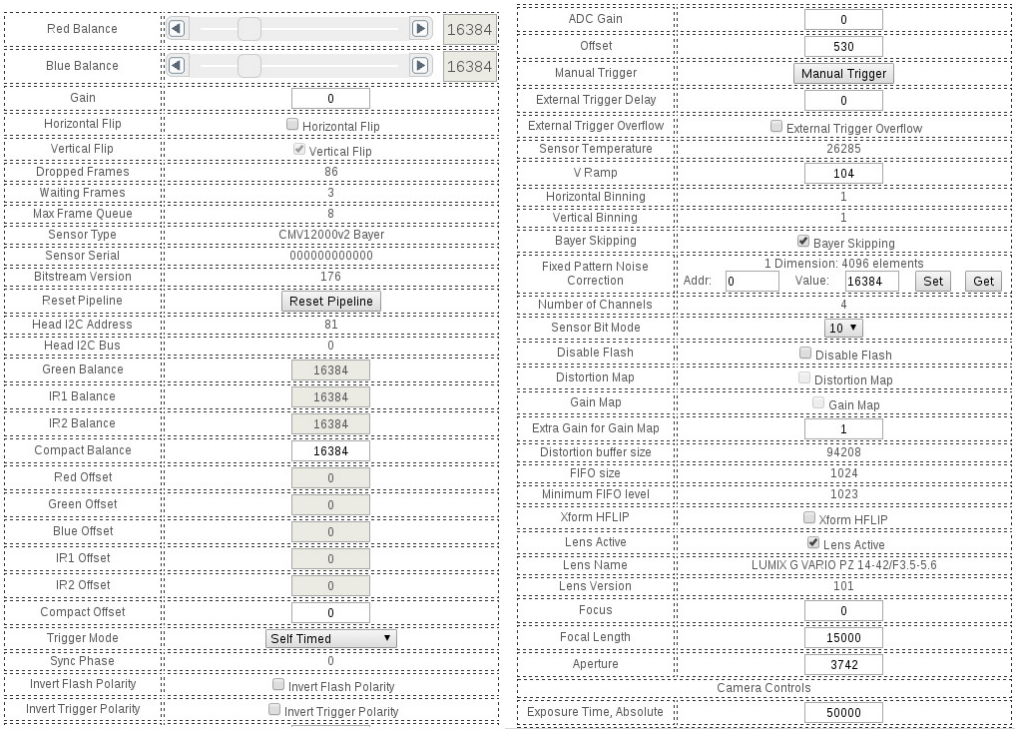


Figure 9.2.: Controls available on a CMOSIS sensor

controls on a camera with a CMOSIS head. Video4Linux2 provides a library (v4l2-ctrl) for handling those controls and expose them to the user through the Video2Linux API. The different controls have flags defining their behaviour, like READ_ONLY, or WRITE_ONLY.

v4l2-ctrl implements an intelligent cache system that reduces the expensive iomem operations with the hardware. On the original design, the cache system was not compatible with error flags. As a result of this Thesis a new flag was added to the v4l2-ctrl framework that support such controls. This new flag, V4L2_CTRL_FLAG_EXECUTE_ON_WRITE, has been merged into the main Kernel branch.

9.1.3.2 *Xform Module*

As described on the chapter 7, the Xform functionality is configured with two files: i.e. gain and distortion, as configuration.

The Xform Module is modeled as a standard Video4Linux2 output device with two ouputs: V4L2_PIX_FMT_GREY for the gain image, and V4L2_PIX_FMT_QTEC_DISTORTION for the distortion image. The format of V4L2_PIX_FMT_QTEC_DISTORTION is detailed in the following code listing.

```
struct dist_pixel_data{
        uint8_t cy;
        uint8_t cx;
        uint8_t gain;
        uint8_t move;
};
```

Like in the Main Driver, the buffer API is implemented making use of the VB2 library. During the development of the module, some bugs were found in the video output VB2 headers. The fixes were contributed back to the Kernel and subsequently accepted.

9.1.3.3 *Other V4Linux2 Modules*

The rest of the FPGA cores are controlled with Video4Linux2 subdevices. There is one module per subdevice, and they are only loaded if the correspondent hardware is present on the running bitstream, i.e. the CMOS module is not loaded with the CCD head.

QTEC_CCD This subdevice controls the Frame Generator core on the CCD bitstream. The hardware design allows different exposure settings per channel. A custom IOCTL has been implemented to support multiple WOIs. This IOCTL is being superseded by a new API that emulates the per-frame setting of Android HAL v3 [All].

QTEC_CMOSIS This subdevice controls the Frame Generator core on the CMOSIS bitstream. It has the same API as the CCD Frame Generator

to support multiple WOIs. The different trigger methods, i.e. self timed or external trigger, are selected with a v4l2-ctrl.

QTEC_ROIC The InGaAs and microbolometer heads share the same bitstream, this is due to the similarities in the readout electronics. This subdevice controls the Frame generator present in that bitstream. The main quirk of this subdevice is its interface with the Dead Pixels Flash, located on the sensor board.

QTEC_WHITE This subdevice controls the White Balance core present in all the bitstreams. It exports the gain and offset parameters for each channel via v4l2-ctrls.

QTEC_M43 This subdevice is in charge of controlling the MicroFour-Thirds lens. There are v4l2-ctrls for the Lens Type, Focus, Focal Length and Aperture. Another control is used to turn off and on the lens. This just represents a subset of what MicroFourThirds is capable of, but enough for most of the applications.

9.1.4 *Serial communication*

Although the serial port is almost obsolete in the consumer electronics, it is still in use in the industrial and research sectors. Among others, the serial port is still in use to control Programmable Logic Controllers (PLCs), motor units and custom scientific components; and also to monitor and debug custom machines. The port that that was commonly found on the 90's computers implemented a point to point single ended protocol denominated RS-232, the industrial variant, which is multi point and dual ended is denominated RS485 [YZD08].

On Linux, there was a specific set of IOCTLs (TIOCGRS485 and TIOCSRS485) for controlling the RS485 configuration of a serial port. Those IOCTLs were implemented by only a small subset of devices with a lot of code duplication. The serial device on the camera, although being quite standard, did not have support for those IOCTLs.

As a result of this Thesis, the serial port infrastructure was modified to support natively the RS485 IOCTLs, this way all the duplicated code was removed from the device drivers; and a new module for the serial port on the platform, Fintek F81216A LPC, with support for the RS485 IOCTLs has been developed and merged into upstream. Another minor contribution in this field has been the modification of the Xilinx UartLite driver to build correctly in other platforms than ARM or PowerPC.

9.1.5 *SPI*

As seen on the Bridge Driver section, the FPGA reads is configuration from a Flash, more specifically an SPI Flash. This happens when the system starts and during reconfiguration. Before the FPGA is reconfigured, the CPU must verify that the content of the FPGA matches the type of head used. Because of this, the SPI read / write speed will determine the boot time, which is a critical feature on an embedded product.

The original driver for the Xilinx SPI core, present in the Linux Kernel, required 118 seconds to read the Flash and produced more than 83000 IRQs in the process. The poor performance was due to a suboptimal access to the core registers and a missuse of the IRQs. Also, the module did not support all the SPI modes supported by the core.

As a contribution of this Thesis, the module was modified to improve its performance, achieving a whole flash readout in 41 seconds (2.8x faster), with only 100 IRQs (830x less). The module was also enhanced with support for all the modes required by all the used sensors. All these changes were contributed and merged into the Kernel.

On the following paragraphs, the contribution of this Thesis regarding the Xilinx SPI module are detailed:

9.1.5.1 *IO access*

The core has a size-programmable buffer with a set of flags to determine its occupancy level.

On the original implementation, the buffer was filled following this algorithm that requires two IO access per SPI word:

```
while (NOT IOREAD(buffer_full))
    IOWRITE(buffer_data, buffer[i++]);
```

On the new implementation the buffer size is pre-calculated at boot time using the following algorithm:

```
BUFFER_SIZE = 0;
IOWRITE(stop_transfer);
IOWRITE(reset_buffer);
while (NOT IOREAD(buffer_full)){
    IOWRITE(buffer_data, 0);
    BUFFER_SIZE ++;
}
IOWRITE(reset_buffer);
```

Once the buffer size is known, the buffer can be filled with only one IO access per byte:

```
for (i=0; i < BUFFER_SIZE; i++)
    IOWRITE(buffer_data, buffer[i++]);
```

9.1.5.2 *IRQ usage*

On microcomputer architectures, the CPU is accompanied with peripherals that perform tasks commanded by the CPU. The CPU can poll the peripherals to find out when the task is finished or get an IRQ from the device when the task is completed. Although IRQs allow multitasking there is a overhead associated with its use: The IRQ needs to be acknowledged and processed, the CPU will suffer a context switch and the latency of the whole system will be affected. If the processing time of the peripheral task and the overhead time of the IRQ handling are known before hand, the CPU can choose efficiently between IRQ or polling.

On the previous implementation, the module did only support IRQ transfers. Each SPI transfer required an IRQ and 6 extra IO accesses. Small transactions were heavily penalized with this access method.

On the new implementation, IRQs are only used when the transaction has at least the same size than the buffer.

9.1.5.3 *SPI modes*

On SPI a master device communicates with a slave device using 4 wires, clock, Master to Slave, Slave to Master and Chip Select. The standard supports multiple configurations of Clock and Chip Select Polarity, Data Phase and Data order.

On the original module implementation, only the Clock Polarity and the Data Phase could be configured, leaving unsupported a great amount of SPI devices.

On the new implementation, the module was extended to support configurable Chip Select Polarity, Data Loopback and LSB First data order. This change was also contributed to upstream and merged.

9.1.6 *USB Gadget*

One of the peripherals available for the platform is a USB Peripheral Controller. Thanks to this peripheral, the platform can be connected to another system emulating the behaviour of any USB device: e.g. Mass Storage Device, webcam, printer...

At the time of writing this Thesis, there is only one PCIe USB Peripheral Controller with support for USB 3.0: the Avago USB338x[plx].

This device had a very poor Linux support based on a module from its manufacturer, Avago, which was not integrated in the Kernel and was incompatible with the latest versions. As a result of this Thesis, this driver was refactored and contributed to the Kernel. This work has been reused for other companies, like Linaro and Intel, that have also shared their progress with this chip.

9.1.6.1 *GPIO and LEDs*

Chapter 8 describes how the GPIOs are used in the form of LEDs in the camera to show its status. These GPIOs are controlled by a Xilinx core in the FPGA.

The following paragraphs detail the changes contributed to the Linux kernel in relationship with the GPIO and LED subsystems.

9.1.6.2 *Xilinx GPIO module*

The legacy implementation of the GPIO module in the kernel presented a series of problems that prevented its use on the platform.

First of all, it could only be used on ARM and PowerPC platform, this could be solved modifying the Kernel build configuration file [GR03].

Then, the original module did not support hot-plug enumeration. Because of this, it could only take control of the GPIOs cores enumerated before loading the module, but on the QT5022, the GPIOs are enumerated at run time. This problem was fixed porting the driver to the Platform Device Infrastructure.

Finally, the module did not handle properly the resources on cores configured in Dual mode, i.e. in dual mode a single GPIO core can control 64 GPIOs instead of 32. This was fixed following the directions of the GPIO tree Maintainer.

9.1.6.3 *GPIO core*

On the Kernel, the GPIO drivers that had memory mapped resources were handled with a set of helper functions called gpiochip-mm. These helpers were developed considering that the GPIOs could not be removed, but on the proposed platform, when the FPGA is reconfigured, all the cores from the previous design are removed.

Gpiochip-mm was extended to support GPIO removal. This modification required modifying all the modules using the gpiochip-mm functions, that now can support GPIO removal. All this changes were contributed back to the kernel and accepted.

9.1.6.4 *LEDs*

On the Linux Kernel, LEDs are controlled by the LED class, that abstracts all the access to the LEDs and provide the concept of trigger: A trigger is an event generated by any module that changes the status of an LED, e.g: heartbeat, hard-drive status, network-card status, etc.

On the LED class, each LED has an unique name, this name is defined statically or by the Device Tree file. In systems with a single static Device Tree is easy to check that all the LEDs have different names, but on systems with a dynamic Device Tree this condition can not be taken for granted. The LED class interface has been modified to support LEDs with the same name. This change has been contributed and accepted.

9.1.7 *Other contributions*

Through the development of this Thesis there has been more contributions to the kernel than the previously described:

On Linux, the different modules can be compiled as part of the main Kernel file or as separated files. When they are compiled as separated modules, a userland applications loads the required modules based the hardware present on the system. Each module has a list of hardware that it supports and the kernel creates a dynamic list with the hardware enumerated on the Kernel. For PCI devices, there was a bug on the way the list of devices was created that avoided loading certain modules. This bug has been present in the driver core for more than 9 years. As a result for this Thesis, a patch has been contributed to the community which decided to accept it and backported it to previous versions of the kernel.

A different LED controller (PCA9634) was used on a preliminary version of the platform. This chip was not originally supported by the Kernel. A new driver was developed for this controller, and later on shared and merged into the main version of the kernel.

9.2 NON-VOLATILE MEMORY PROVISIONING

The current implementation has a great number devices with non volatile memories. The methodology for provisioning these devices is described in this section.

9.2.1 *BIOS*

The BIOS is the first code executed from the CPU, this code is in charge of initializing the different peripherals in the system and is loaded from a non volatile memory The current implementation loads the BIOS from a SST flash chip "SST25VF032B" with 4096 KiB.

The content of the BIOS is based on a closed-source Phoenix reference code customized by a third company. An open source version of the BIOS denominated coreboot can also be used for debugging, but it is far for being ready for production on the selected APU.

Through the development process of the camera, different version of the Phoenix and coreboot BIOS were loaded into the flash, this was done with Flashrom [fla15]. Flashrom is an open source utility with support for more than 450 different chips and 280 chipsets. Both the chipset and the flash were supported by flashrom, so adding support for the camera was a simple task.

9.2.2 *Ethernet Card*

In this design, as specified on the chapter 8, the network interface consist on an Intel 82580 chipset. At boot time, the chipset reads its configuration, including the Media Access Control (MAC) address and link speed, from an EEPROM.

Linux provides functionality to change on runtime the MAC address and the link speed, but it does not provide an interface to change the configuration EEPROM of the Ethernet Card. Although it is possible to find some tools to change the MAC address permanently, there is no tool to change the whole EEPROM configuration.

Instead of making a new tool, Flashrom has been modified to support Intel Network Card EEPROMs. This change is the first EEPROMs support for flashrom and was finally merged into the main version after months of peer review.

9.2.3 *FPGA Flash*

On this implementation, the FPGA reads its configuration from an SPI Flash.

Xilinx tools provide support for programming the FPGA through the Joint Test Action Group (JTAG) port, but the smallest version of the tools, denominated Xilinx Lab Tools requires 900 MiB of space, have a very restrictive license and a lot of dependencies.

An Open Source alternative to the Xilinx Lab Tools is xc3sprog[xc315]. Xc3sprog can program Xilinx FPGAs through JTAG cables. Xc3sprog can be easily (and legally) distributed, requiring less than 1 MiB of hard drive space and very little dependencies.

As a result of this Thesis, some patches were shared with the community to support the Flash format for the Xilinx Kintex FPGAs.

9.2.4 *Flash Unit*

As described on the chapter 8, the flash unit is composed by one main microcontroller, P18F87K22 [p1811], connected to the Processing Unit through an USB to serial converter and 8 PIC16F1823 [p1612] connected to the main microcontroller.

The microcontroller needs to be reprogrammed on the fly to perform advanced flash patterns.

On the main microcontroller, the reprogramming capabilities are achieved with a modified version of the serial port bootloader: tinybootloader [tin15]. A custom program has been developed for the main CPU that can reprogram the microcontroller.

On the small microcontrollers, due to the small size of their flash there is no bootloader. Instead, the main microcontroller has an implementa-

tion of the Low Voltage PIC Programming protocol. Through the Low Voltage Programming protocol all the memory regions of the small PICs can be reprogrammed. As with the previous case, a custom program has been developed on the main CPU for communicating with the microcontroller and pushing the new code.

9.2.5 *Micro Four Thirds Module*

The Micro Four Thirds Module follows a similar logic than the Flash Unit, it is a microcontroller that needs to be reprogrammed by the main CPU. The main differences are that the communication to the microcontroller is done via SMBUS and that the chip is from a different family, specifically an AVR ATMEGA328P [avr14].

In the literature is has been impossible to find an SMBUS bootloader for this device, so it has been developed from scratch with a userland tool.

9.2.6 *Hardware Monitor*

The Hardware Monitor is also governed by a microcontroller (MicroChip PIC18F26K2 [p1811]) that needs to be provisioned.

On standard configurations, the first program executed in the device is the bootloader, that jumps into the application after a timeout of inactivity, usually between 1 and 10 seconds.

In this particular case, the main application needs to be ready within 100 milliseconds after powerup. So the boot order has been swapped: The main application will jump to the bootloader on demand.

As with the previous case, the bootloader and the userland tool have been developed from scratch.

9.3 COMPUTER VISION STACK

The main goal for the platform is to be able to use it in as many applications as possible. On the hardware side, this has been achieved with a

modular design, on the software side this is achieved with a big selection Image Processing Libraries.

As previously described, the interface to the imaging sensor/FPGA is done through the Video4Linux2 interface from the Kernel. The images delivered by Video4Linux2 have been pre-processed by the hardware with an imperceptible latency of a couple of hundreds of lines.

It is worth enumerating the image processing steps performed on the Image Processing Pipeline, described in detail in chapter 7:

- Debayer
- FPN correction
- Binning
- White Balance
- Perspective Correction
- Rotation
- Illumination Compensation
- Lens Calibration
- Pixel Formatting

These local algorithms need to be complemented by other computer vision algorithms, like segmentation or classification.

In this section, the computer vision libraries embedded in the platform will be described, with a final comparative of all them. This comparative can be used as guideline for the final user. Figure 9.3 gives an overview of the different video processing stacks available on the system.

9.3.1 *Video4Linux2 Utils - libv4l2*

Video4Linux2 supports a great amount of pixel formats and readout modes but the imaging sensors only implement a subset of them.

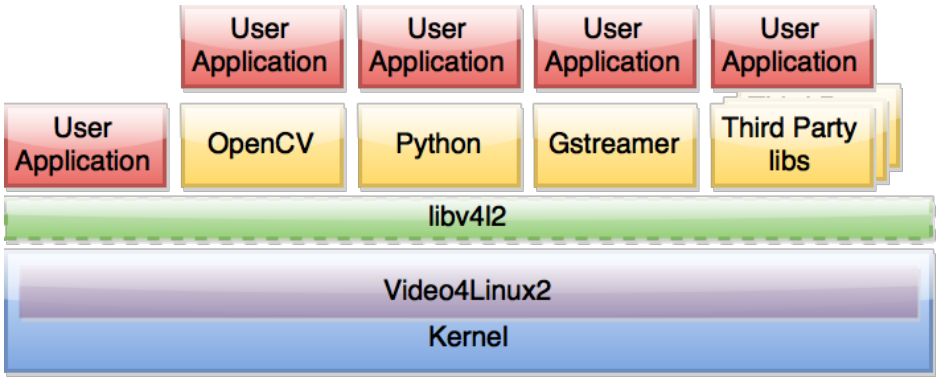


Figure 9.3.: Video Processing Stack

Lib4l2 is build as an adaptation layer around Video4linux that automatically converts the frames from any sensor into the required format by the application.

This way, any Video4Linux2 application can work on all the Video4Linux2 hardware regardless of its supported pixel formats or readout modes. Lib4vl2 can be explicitly used by an application or it can be dynamically preloaded to hijack the calls to Video4linux IOCTLs.

Lib4l2 is developed and maintained by the developers of Video4linux, and i is distributed with a set of tools for testing and configuring Video4Linux2 devices:

- v4l2-compliance: Runs a series of tests to verify the compliance of a driver to the Video4Linux2 standard.
- v4l2-ctl: Console tool to configure and capture images from a driver.
- qv4l2: Graphical version of v4l2-ctl, Featured on Figure 9.4

As a result of this Thesis, libv4l was extended to support V4L2_PIX_FMT_Y16, V4L2_PIX_FMT_Y16_BE, V4L2_PIX_FMT_BGR32 and V4L2_PIX_FMT_RGB32 formats. qt4l2 was also extended to support the formats V4L2_PIX_FMT_Y16 and V4L2_PIX_FMT_Y16_BE. All these changes were contributed and accepted in the main distribution of libv4l2.

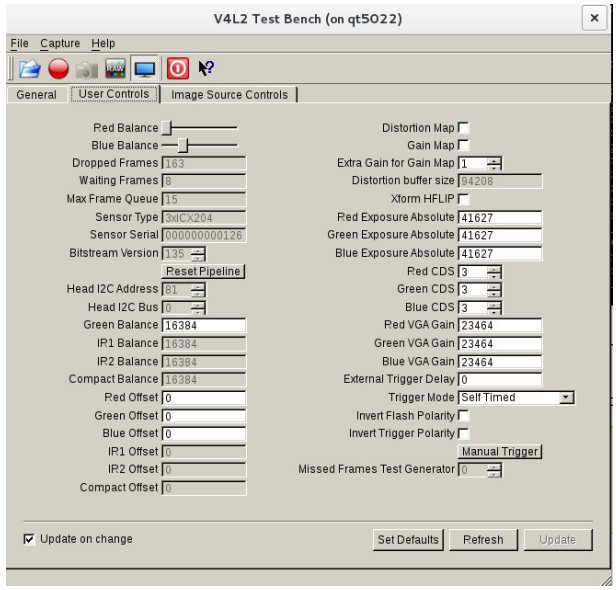


Figure 9.4.: Qv4l2 running on a 3 CCD camera

9.3.2 *OpenCV*

Open Source Computer Vision Library (OpenCV) [B+00] is a library with implementation of several hundreds of computer graphic algorithms. After years of active development, it has become the de facto standard in computer vision with many research papers providing OpenCV implementations of their algorithms [FZWL12, SYJ14]. Although the current version is mainly developed in C++, the library provides bindings for C, C++, Java and Python.

The original OpenCV committee was formed by academic members from Stanford University, UC Berkeley, Caltech, Boston University and Duke University. Later on, it has also received contributions from the Industry: Intel, Google, AMD, NVIDIA, Willow Garage,Phase Space...

OpenCV has native support for Video4Linux2 devices and some of their algorithms can make automatic use of OpenCL in the presence of supported hardware [Gas14].

The availability of its source code and the flexibility of its licence simplify its deployment in Industrial and research environments.

9.3.3 *Python*

Python is an interpreted language that has won popularity through the last years thanks to its minimalistic syntax, and a great availability of libraries.

With the appropriate libraries it can be used as a powerful scientific and computer vision tool [Sol12]:

1. ctypes: Data type library compatible with C. It can be used to interface the kernel API
2. NumPy: N-dimensional array object with linear algebra, Fourier transform, and random number capabilities [Dev13].
3. Bohrium: Runtime environment that vectorices numpy objects with Hardware Accelerators such as OpenCL [KLB+13].
4. SciPy: Numerical routines for numpy objects[JOP14].
5. Matplotlib: 2D and 3D plotting library [HD07].
6. Ipython: Interactive shell for python with support for Matplotlib figures [PG07].

Figure 9.5 shows an ipython session that combines the previously mention libraries, constituting a great alternative to proprietary tools like Matlab or Mathematica.

9.3.4 *GStreamer*

GStreamer [TBW+01] is an Open Source library for building multimedia applications. In GStreamer, an application is a graph of components that process a flow of video, sound or any other BLOB. The library is

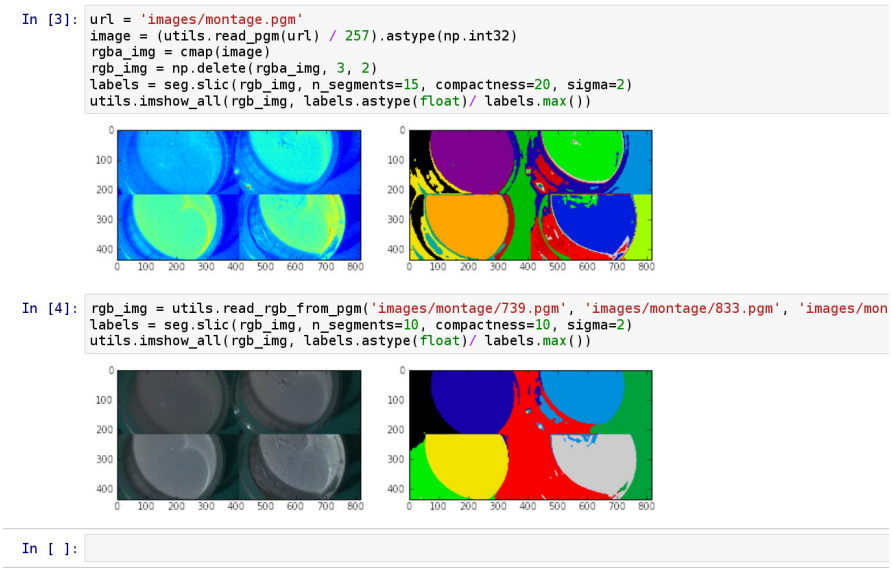


Figure 9.5.: Computer Vision notebook on Ipython

designed to have a very little overhead and provide efficient ways of pipelining and debugging.

GStreamer is distributed with over 250 plugins and 1000 elements capable of handling different protocols, inputs (including video4linux2), formats, codecs, filters and sinks (Figure 9.6). GStreamer is being developed by a very active community.

Applications are easily developed by connecting elements, if there is no element available for an specific task, the library can be extended with custom-made plugins. This development model facilitates the reusability of code.

Although the most common user case for GStreamer is building multimedia applications, it can also be used in Computer Vision Applications [BD11]. It is expected that the number of plugins in the area of computer Vision will increase in the near future.

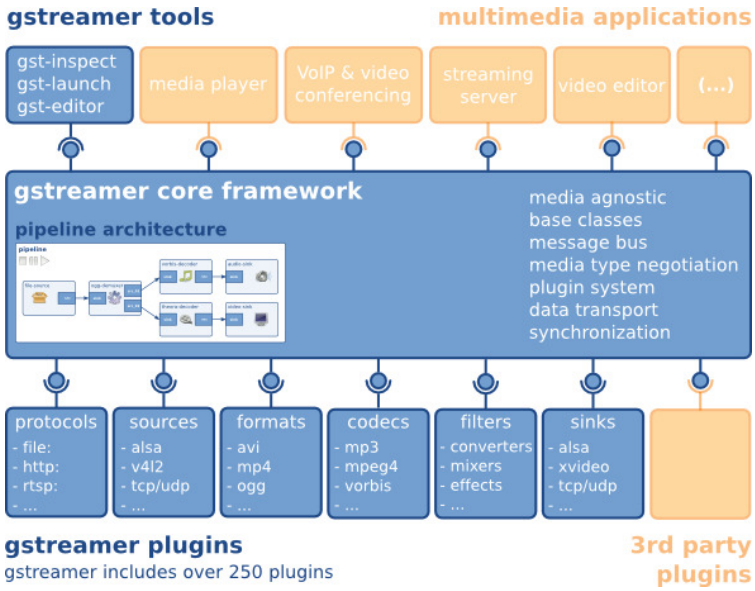


Figure 9.6.: Gstreamer Overview

9.3.5 *Third Party libraries*

Because the platform is based on a X86 processor that runs standard a Linux Kernel, there is a great availability of Proprietary Computer Vision Libraries that can be used.

These libraries are generally distributed in binary form, i.e. no source code available, hindering deep debugging methodologies [Jan12].

The Proprietary Libraries MVTEC Halcon [Ver12] and Cognex Sentisight [sen15] have been tested on the camera with satisfactory results. Halcon works out of the box with the platform due to its support of Video4Linux. Sentisight uses a Gstreamer pipeline for fetching images into the library, with is also supported on the camera.

9.3.6 *Comparison*

This subsection pretends to serve as a small guideline for selecting the right Computer Vision Library for each application and development stage. The user shall decide which is the right library based on:

- Ease of use: How easily the user can make changes and experiment with the camera.
- Reusability: If is possible to reuse code from previous projects, and if the code generated can be reused later on.
- Number of Algorithm: How many Computer vision algorithms are already implemented in the library.
- Performance: Efficiency of the structures and computer vision algorithms
- Video4Linux2: How much of the Video4Linux2 standard is supported.
- Multi Thread: If the library supports multiple threads efficiently.
- OpenCL support: If the application can make use of the Acceleration Unit.
- Documentation: How well documented is the library.
- Support: How active is the community behind the library and how much they are willing to help.

For the libraries distributed with the platform, Table 6 serves as a simplified decision making matrix, based on the previous criteria.

As a general recommendation, it could be concluded that an application should be developed with libv4l2 only if it is required the highest performance possible from the platform. OpenCV is a very balanced platform that can be used in the experimental phase as well as in the product phase. Python is perfect for research and prototypes. Finally,

Table 6.: Comparison of Computer Vision Libraries

	libv4l2	OpenCV	Python	GStreamer
Ease of use	+	++	+++	+
Reusability	+	++	++	+++
# of Algorithms	+	+++	++	+
Performance	+++	++	+	+++
Video4Linux2	+++	+	+	++
Multi Thread	+++	++	+	+++
OpenCL Support	+++	+++	+	+
Documentation	+	++	+++	+
Support	+	++	+++	++

In each field, more + is better

GStreamer has a steeper learning curve that any of the other libraries, and does not have as many computer vision algorithms implemented as OpenCV or Python, but it produces code that can be easily reusable and has an excellent performance.

9.4 OTHER APPLICATIONS AND LIBRARIES

The platform requires from other tools, like debuggers, profilers, network tools, graphical interfaces and browsers, in order to be a standalone data processing system.

Although this software can be assembled manually, it is much more convenient to use a Linux Distribution with thousands of applications already packaged and maintained and with a packaging system that allows a flawless upgrade and deployment methodology.

There are many distributions available for Linux, focused in different fields like Science, High Performance, Desktop Computing or Embedded.

9.4.1 *Linux Distribution: Yocto Project*

Yocto Project is the Distribution used on the proposed implementation of the platform. It is a distribution that is focused on embedded devices, supports multiple architectures, can be easily extended and provide thousands of packages as well as a rich IDE with good debugging capabilities [LSH13, SA14]. Yocto Project has a very rich ecosystem formed by companies like Intel, AMD, Broadcom, Huawei as well as other open source communities like Open Embedded.

Yocto Project has been extended to support the proposed Computer Vision System with the development of a Board Support Package that integrates: specific optimizations for the platform, the modules for the Image Processing Unit, the latest drivers for the APU and discrete GPU, and the custom applications for accessing the non-volatile memories on the system.

As a result of this Thesis, some of these changes were contributed back to Yocto Project in the form of 17 merged patches in different areas:

1. OpenCV: Include support for OpenCV3 and fix the support for OpenCV2.

2. Gstreamer: Include Gstreamer's unit testing platform.

3. Graphical Server: Provide the modules required by AMD proprietary drivers.

4. Packaging system: bug fixes and speed up the creation of package repositories.

5. Toolchain: Fix building on AMD64 platforms.

The Kernel profiler and tracer: perf [dM09], has been modified to support the default location of the symbol files of Yocto. This change has been merged into the Kernel main tree.

As a result of the collaboration with Yocto Project, Qtecnhology was included into Yocto Project as a Participant Organization.

9.5 GRAPHIC DRIVERS

The current implementation of the system has two AMD's GPUs. The interface to these devices is OpenCL, a heterogeneous programming language. In this section the properties of the drivers from AMD are discussed.

Graphical acceleration is a field mined with thousands of patents. Algorithms as simple as a logic XOR are susceptible of being patented [ST80]. As the number of patents grows, computer Graphic engineers must devote a lot of effort analyzing their codes in search for possible violations of third party patents, and implementing workarounds for avoiding them.

In this scenario, the companies avoid disclosing details about their implementations, as they represent opportunities for their competitors to legally compromise them. As a result of this, most of the companies distribute their drivers in binary form, i.e. , with no source code.

This kind of distribution has some issues [AC06]. First of all, it is much more difficult to debug software without its code, and reverse engineer the drivers could be even considered illegal in some countries. Then, the integration with other parts of the system is more complicated. Finally, there is no clear changelog in the releases from the manufacturers.

9.5.1 *AMD Proprietary Display Drivers*

Figure 9.7 shows the structure of AMD's Graphical Stack. The components distributed in binary form are displayed in red, the parts with source code available is represented in green. The yellow blocks are the part of the stack that belong to the OS.

On AMD drivers, all the communication with the graphic card is done through a BLOB on the kernel. This BLOB interfaces with the rest of the kernel through a wrapper denominated fgrlx-public which code is available.

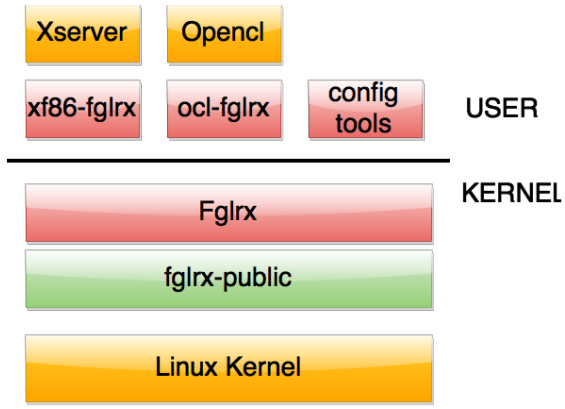


Figure 9.7.: AMD Proprietary Display Drivers

Fgrlx-public takes into account the differences in the internal kernel API, so the same BLOB can be used on any kernel. An example of flgrx-public can be seen on the following snip set:

```
unsigned int kas_try_to_freeze(void)
{
#if LINUX_VERSION_CODE <= KERNEL_VERSION(2,6,10)
    return 0;
#else
#if LINUX_VERSION_CODE <= KERNEL_VERSION(2,6,12)
    return try_to_freeze(PF_FREEZE);
#else
    return try_to_freeze();
#endif
#endif
}
```

Advanced functionality such as V-sync or thermal throttling is configured with proprietary applications that interface the proprietary kernel module.

The other parts of the stack are a Xserver and OpenCL drivers, each of them interact with the Fglrx module through an AMD proprietary interface.

9.5.2 *Benchmark and profile tools*

Because of the closed nature of the AMD driver, it needs to be constantly verified for correctness and performance whenever there is a change on the software stack.

In this system, this is done with the AMD OpenCL samples, generic OpenCL benchmark tools and custom tools.

AMD OpenCL Samples [Sta11] are a set of reference implementation of different algorithms such as Matrix Multiplications or Convolution.

The main generic OpenCL benchmark tool on the literature is Clpeak [Bha14]. It is capable of analyzing the performance of an OpenCL platform in different fields such as single-precision, double-precision, integer, transfer bandwidth and kernel latency among others. As a result of this Thesis a series of contributions were made to Clpeak and accepted into the main distribution. These contributions are in the areas of cross compilation and result output.

The custom tools are standalone versions of the applications described on the chapter 10. These tools verify that future versions of the platform have similar performance.

9.5.3 *Beyond AMD Proprietary Display Driver*

During the last years AMD main competitor, Intel, have firmly believed on the Open Source model: they have created multiple Open Source Centers around the world where thousands of engineers collaborate with the Open Source community to improve the support of their hardware.

Thanks to these efforts, Intel is now represented in most of the Board of Directors of graphic-related projects such as Xorg, Mesa or even the Linux Foundation. Despite of the technical inferiority of Intel's GPUs,

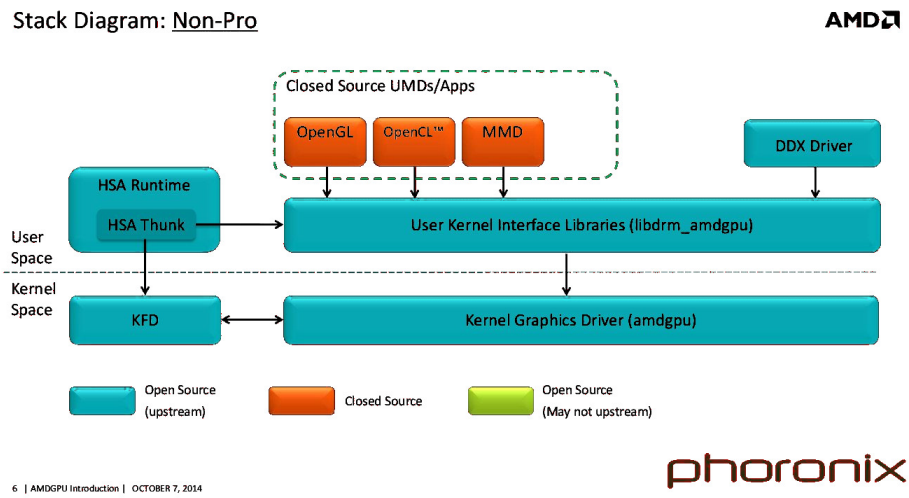


Figure 9.8.: Future AMD Open Source Stack

the great quality of their drivers and integration with the Open Source Stack results in very competitive products.

These results, have made AMD reconsider their approach to Open Source. On the last year AMD have revealed the long term plans for their Graphic Stack (Figure 9.8 from [LT15]). This new drivers consist on an Open Source kernel module that interacts with the driver and two different stacks, an Open Source and a Closed Source, both using the same interface from the Open Source Kernel Module. The Open Source stack follows Intel's design: mesa, and xorg. The Closed Source follows the current AMD model. The user can decide which stack to use.

At the time of writing this Thesis, the Open Source driver has not been completely accepted into the Kernel and it is only targeted for GPUs manufactured from 2015. Only future implementations of the Processing Unit of this platform will benefit from these drivers.

10

APPLICATIONS

On the previous chapters of this Thesis, a novel modular Computer Vision System have been described from multiple perspectives: hierarchy, dataflow, hardware and software level. On this chapter, the proposed system is validated in four real scenarios.

The applications on this chapter have been developed in collaboration with companies in the Food and Packaging industry and Universities. The heterogeneity of the presented applications constitute a great example of the versatility of this system.

Contents

10.1 OPTICAL POTATO GRADER

This application is a commercial product by Newtec A/S, Denmark. The design is an original work from Henrik Andersen and Daniel Bengston (Newtec). The role of the author of this thesis has been the implementation of the low level acquisition layer, profiling and optimizations to obtain the best performance out of the QT5022 platform.

Potatoes play an important role in the diet of millions of people, according to the Food and Agriculture Organization of the United Nations, the world production of potatoes in 2013 was about 368 million tonnes [FAO09].

Before a potato can be industrially processed or distributed to the public, it needs to be classified by size and defective potatoes should be disposed:

- Size grading is done for aesthetic and practical reasons: Perfectly spherical potatoes are considered a premium product with prices up to 20 time higher than the rest of the non spherical potatoes. Furthermore, some processed food products like french fries require a minimum potato length.
- Defects detection is done for aesthetic and health reasons: Supermarkets have high standards of quality and don't accept potatoes with skin problems or albinism. In addition, green potatoes are rich in Solanine, a compound toxic even in small doses [Sla90].

The grading process gives value to one of the cheapest crops to produce.

Newtec, has implemented an Optical Potato Grader [cel15], shown on Figure 8.17, with the QT5022 platform using the following configuration:

- Light: High Speed Fluorescent.
- Lens/Optics: C-Mount Fixed Lens + 2 dichromatic prisms (3 channels).
- Imaging Sensor: 3 x CCD Sony ICX204.

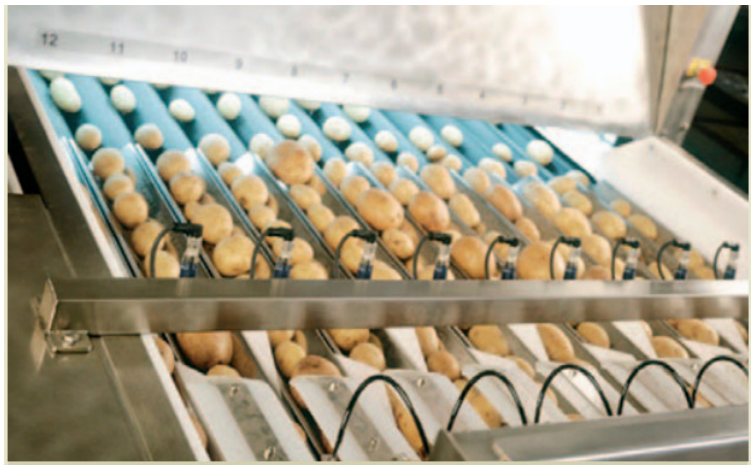


Figure 10.1.: Celox XT Lanes

- Processing Unit: AMD G-T65N APU.
- Acceleration Unit: AMD Radeon E8860.

10.1.1 *Image Acquisition*

Washed potatoes are separated into 12 lanes by a vibration element. The frequency and amplitude of the vibration is configured to avoid product overlapping. Figure 10.1 shows how the products are separated on the infeed section.

The separated products enter the imaging section, where they are rotated in front of three cameras, one camera every 4 lanes. The rotation allows the camera to have a view of the sides of the product. Figure 10.2 shows the view from one of the cameras during calibration. The white element on the figure is a white target used for white balancing.

The border of the machine, i.e. the top of the image, serves as a grey reference, used to control in real time the white balance of the camera, that is affected by the decay of the fluorescent tubes as it ages.

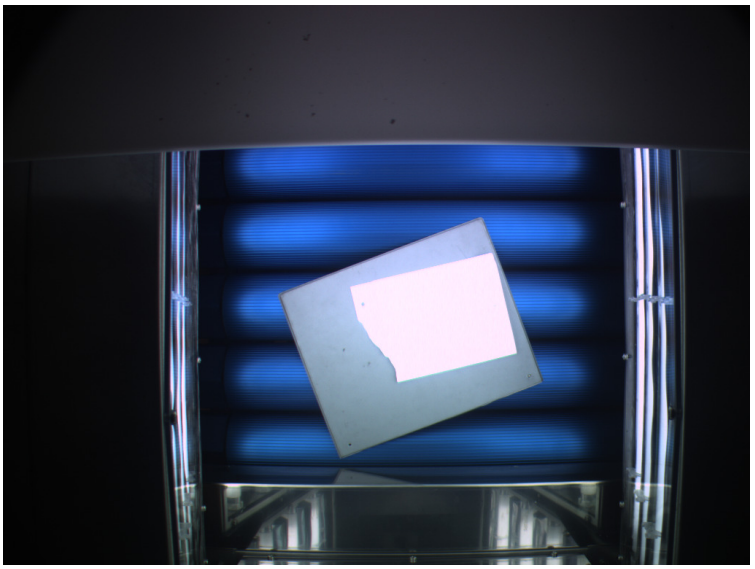


Figure 10.2.: View from one of the camera of the imaging section

The areas with no information are skipped in order to speed-out the readout time. The final image is shown on figure 10.3: The black sections and the green lines are added for visualization purposes. In this application the skipping allows an increase of 33% of framerate.

10.1.2 *Multilayer Perceptron*

The main imaging algorithm executed on the camera is a Multilayer Perceptron, a form of neural network. The selected architecture has 8 input neurons and a hidden neuron (Figure 10.4). This particular architecture has been used successfully by Newtec during many years for classification and segmentation of products in different machines.

Every output pixel uses 8 input pixels, this is: output 1 is obtained from input pixels [1,2,3,4,5,6,7,8], output 2 from input pixels [2,3,4,5,6,7,8,9]... Each output pixel requires 216 multiplications.

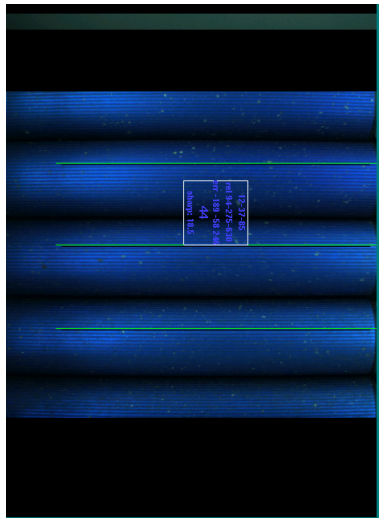


Figure 10.3.: Image after skipping

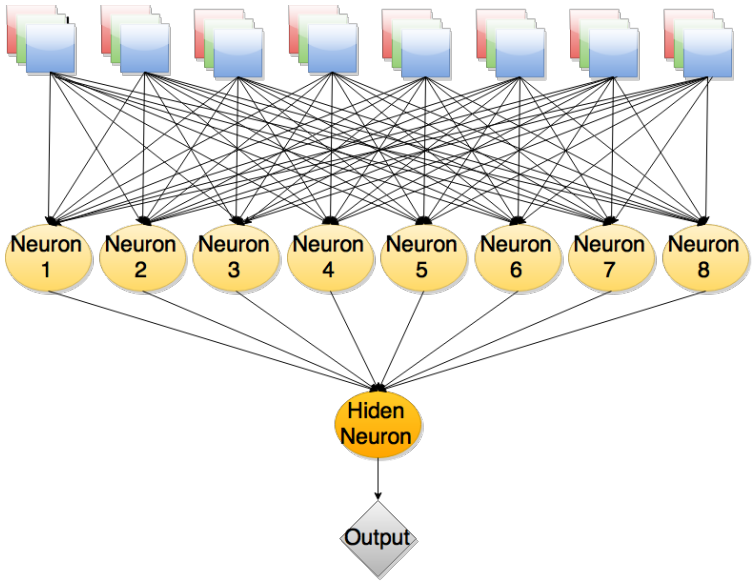


Figure 10.4.: Neural Network Architecture

This typology of data processing, i.e. symmetric processing on all the pixels and no dataflow instructions, takes great advantage of the GPU architecture.

10.1.3 *Imaging algorithm*

Figure 10.5 shows all the steps in the product processing.

1. The potato is segmented from the background with the Multilayer Perceptron described in the previous section. The result is a gray-scale image where the potato has higher values than the background.
2. A threshold filter is applied to the grayscale image. The white pixels are the potato, the black pixels the background (second image).
3. The properties for each line is analyzed: number of white pixels, location of first and last pixels. This results into a simplified potato (yellow image), that is used as input to the tracking and sizing algorithm.
4. Eight different Multilayer Perceptrons are used to analyze the surface of the potato in search of defects (the last 8 stripes). The operator can choose which filters should use for each potato variety or season.
5. The information from the sizing algorithm and the surface analysis is used to classify the potato in 13 different groups.

10.1.4 *Data Flow*

This section gives an overview of the path the data follows through the system. It is important to notice that most of the steps on this list are pipelined:.

1. OpenCL allocates N buffers for images.

Figure 10.5.: Segmentation and Classification Output

2. The buffer is shared with Video4Linux2 using the USERPTR interface.

3. The Video4linux2 driver delivers a frame in the buffer.

4. The frame is processed with an OpenCL kernel on the APU. Since the buffer has been allocated by OpenCL and the CPU and the GPU share the memory the buffer is processed in place.

5. OpenCV is used on the buffer to perform Thresholding and other Morphology operations.

6. The buffer is copied to the discrete GPU.

7. The frame is processed by 8 OpenCL kernels on the Discrete GPU (Acceleration Unit).

8. The result image is copied to the CPU memory.

9. The CPU merges the results from the surface and size analysis.

10.1.5 *Results*

The Celox XT is capable of classifying up to 28 tons of potatoes per hour and each potato is analyzed individually. The accuracy in sizing is one millimeter, and the whole surface is inspected.

Due to the great flexibility of its 8 configurable filters, it is currently used in more than 200 installations around the world.

The vision platform works at 40 frames per second with a spatial resolution of less than 1 mm^2 per pixel. This resolution allows the machine to find a great variety of defects, such as Grey Damage, Dry Cuts, Rottenness, Fresh Cuts, Black Spots, Green, Silver Scurf, Standard Scurf and indications of Hollow Heart.

10.2 BATCH ANALYZER

This application is a commercial product by Newtec A/S, Denmark. The design is an original work from Henrik Andersen and Albert Antony (Newtec). The role of the author of this thesis has been the implementation the low level acquisition layer and working on the profiling and optimizations stages of the project for getting the best performance from the QT5022 platform.

The potato processing industry delights us with their products the whole year: french fries, potato powder, potato salads. But the mother nature only provides potatoes at the right season, therefore the manufactures need to store them in huge silos.

A good administration of the silos is crucial to obtain the most benefit out of the raw material. The manufacturers should process the final product that best fits the specification of the potatoes at the bottom of the silo: E.g. There is no point in producing long french fries if the bottom of the silo only has small potatoes.

This silo accounting was usually done by manual sampling a small subset of potatoes as they were load into the silo, but this sampling methodology can lead to errors between 6% to 8% [sco15].

Newtec has implemented a machine for helping producers and manufacturers to analyze their batches. Newtec Scout [sco15], shown on figure 10.6, is based on the QT5022 platform with the following configuration:

- Light: LEDs using the Light Module.
- Lens/Optics: C-Mount Fixed Lens.

Figure 10.6.: Newtec SCOUT - 3D Sensor Camera

- Imaging Sensor: 1 x CCD Sony ICX204 Bayer.
- Processing Unit: AMD G-T65N APU.
- Acceleration Unit: AMD Radeon E8860.

10.2.1 *3D reconstruction*

The core of this application is the 3D reconstruction of the potatoes. This is done with a single camera and no help from other optical elements like a mirror.

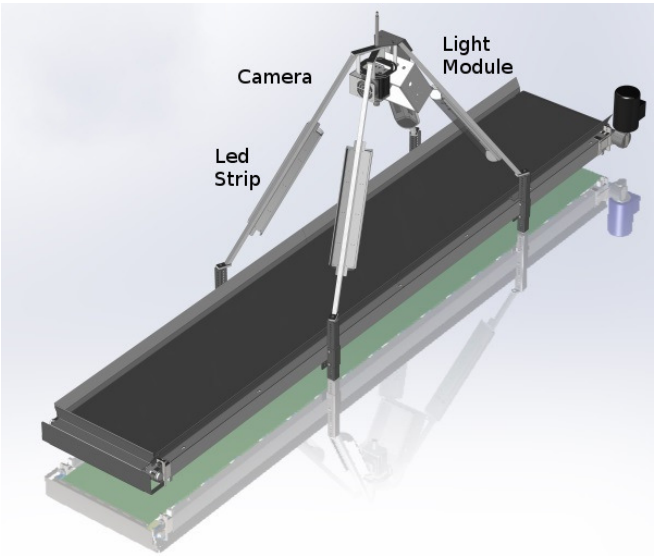


Figure 10.7.: Diagram of Newtec SCOUT

Figure 10.7 shows the location of camera and other components of the machine without the light-isolation tent.

The technique used to create the heightmap of the image is based on the relative speed of the product in front of the camera. Figure 10.8 shows a simplified diagram of the algorithm: On the figure, the conveyor belt moves at a constant speed: for the camera the green product appears to move at a higher speed than the red product, because of the viewing angle of the camera, displayed in pink.

10.2.2 *Data Flow*

This section gives an overview of the path the data follows through the system. As on the previous application, these steps are pipelined.

1. OpenCL allocates N buffers for images

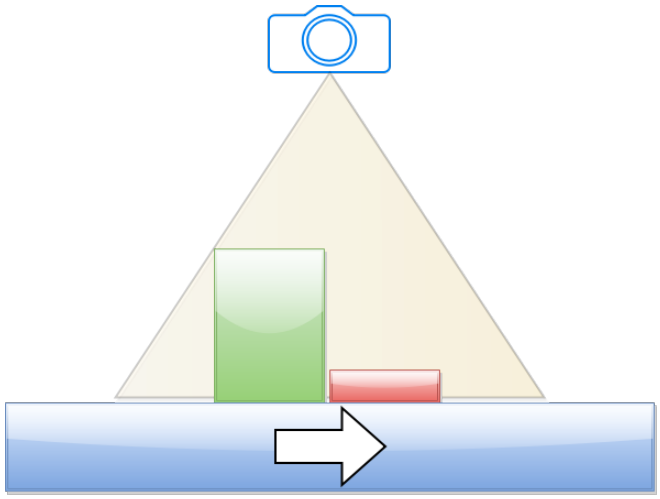


Figure 10.8.: Height Analysis

2. The buffer is shared with Video4Linux2 using the USERPTR interface
3. The Video4linux2 driver delivers a frame in the buffer
4. The frame is copied to the Discrete GPU (Acceleration Unit), using the DMA engine of the GPU.
5. An OpenCL core searches the frame for good reference points.
6. The results are delivered back to the CPU memory.
7. The CPU analyzes the location of the the reference points and compares them with previous images; estimating the traveling speed of those points.
8. The relative traveling speed is used to create a heightmap of the image.
9. The heightmap and the original RGB image is used to create a 3D model of the product.

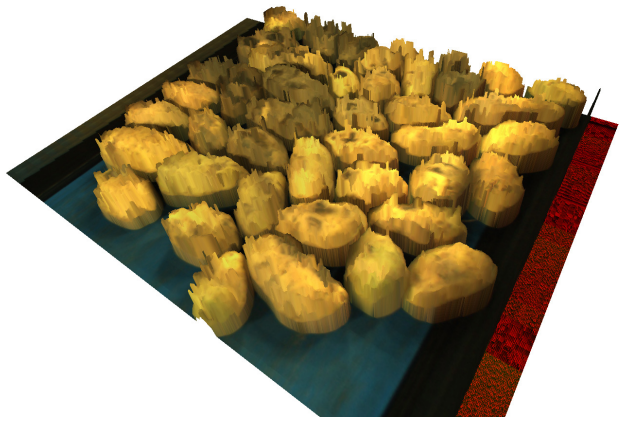


Figure 10.9.: Output of the Batch Analyzer

10.2.3 *Results*

Newtec Scout can be integrated on any production line with a fixed speed conveyor belt. The accuracy of the heightmap will be determined by the speed of the belt.

Although this project was initially designed for Food Processing companies, there is an on-going project to introduce it into potato harvesters. The purpose of this project is to rationalize the use of insecticides and fertilizers to the areas that need it the most.

The camera is working at a speed of 30 frames per second with a resolution at the base of the conveyor belt of less than 1 mm^2 per pixel.

Figure 10.9, shows the output of the Batch Analyzer. This 3D model is obtained in real-time as the products move in front of the camera.

10.3 CHECKWEIGHER

This application is a commercial product by Newtec A/S, Denmark. The design is an original work from Simon Falsig, Martin Hejnfelt and Marianna Buschle. The role of the author has been the implementation of the low level acquisition layer, and working on profiling and optimizations stages of the project, in order to get the best performance from the QT5022 platform.

During the last years, the regulation authorities have specified and enforced very strict rules for customer protection regarding the correct weighing of products sold to the public [Dir04].

Latest regulations enforces a second weight of each product before it is sold to the final customer. This regulation is implemented by adding extra machinery on the production line, with new challenges for the manufacturer:

- Takes space from the, already scarce, production line.
- Could be a bottleneck on the production line.
- It is an extra cost for the producer.

Newtec QC90SV Checkpoint-series [che15], Figure 10.10, is a checkweigher with vision capabilities based on the QT5022 platform with the following configuration:

- Light: LEDs using Light Module
- Lens/Optics: C-Mount Fixed Lens.
- Imaging Sensor: 1 x 4Mpix CMOSIS Bayer sensor.
- Processing Unit: AMD G-T65N APU.
- Acceleration Unit: Not required.

Figure 10.10.: QC90SV Checkpoint-series

10.3.1 *Bag detection*

The main task of the Computer Vision System on the Checkweigher is the bag detection.

The algorithm used for this tasks is based on neural networks, with an architecture similar to the described in figure 10.4. In this product, the major challenge was the training of the network due to the great variety of bags: small, big, transparent, opaque... On the current implementation, the training can be done by the client, who can be guided remotely.

Figure 10.11 shows the results of the algorithm with transparent bags, one of the most difficult product to work with, due to its similarity with the background belt.

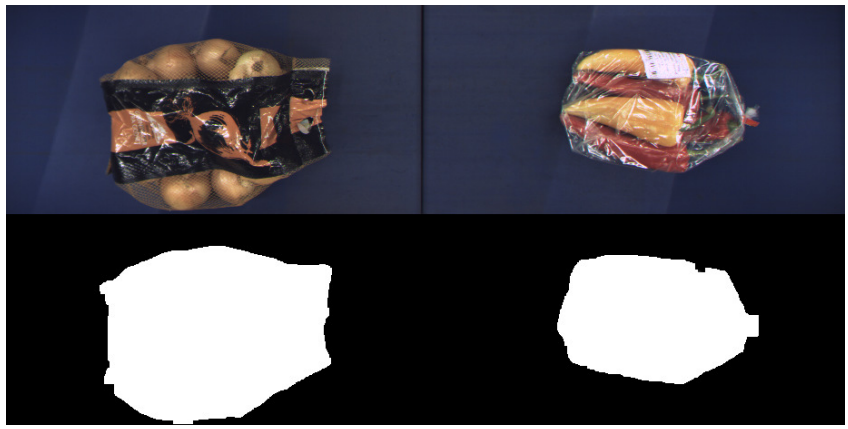


Figure 10.11.: Bag detection algorithm

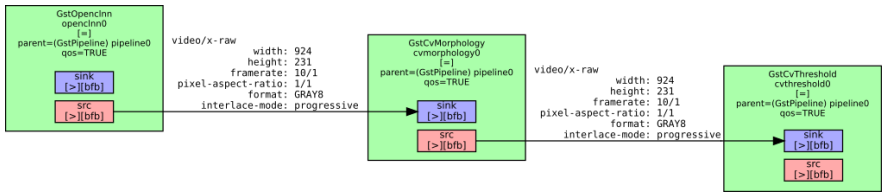


Figure 10.12.: GStreamer pipeline

10.3.2 *Data Flow*

In this case, the steps are pipelined using the GStreamer infrastructure. An excerpt of the GStreamer pipeline is shown in the Figure 10.12

1. Video4Linux allocates N buffers for images.
2. The buffer is shared with GStreamer using the Memory mapping (mmap) interface.
3. The Video4linux2 driver delivers a frame in the buffer.
4. The frame is analyzed by a Neural Network kernel on the APU.

5. The result from the Neural Networks is processed with a Morphology filter to improve the shape of the product.

6. A threshold filter determines what is bag, and what is background.

7. The belt is divided in two areas with different motors. The speed of the second belt is modified, if needed, to fulfill the requirements of the weighing cell.

8. The product is weighed and the shape is analyzed.

9. The color image, weight and shape of the bag determines if the product is removed from the line.

10.3.3 *Results*

The use of Vision on a Checkweigher not only increments the throughput of the system but also enables an extra level of quality control.

Newtec QC90SV Checkpoint-series is capable of weighing up to 90 products per minute and perform the following checks:

- Correct sealing: Check if a product has been damaged by the sealer, or left outside the bag.
- Correct bag: check if the bag used is the right one for the product.
- Correct label: check if the label and add-ons (promo stickers) are placed properly.

The use of the QT5022 has resulted in a great asset for this particular product: Thanks to its rich Open Source Stack, it can be easily tailored to the specific needs and requirements of the application.

10.4 HYPERSPECTRAL CAMERA

This commercial product has been implemented by Qtechnology, Denmark.

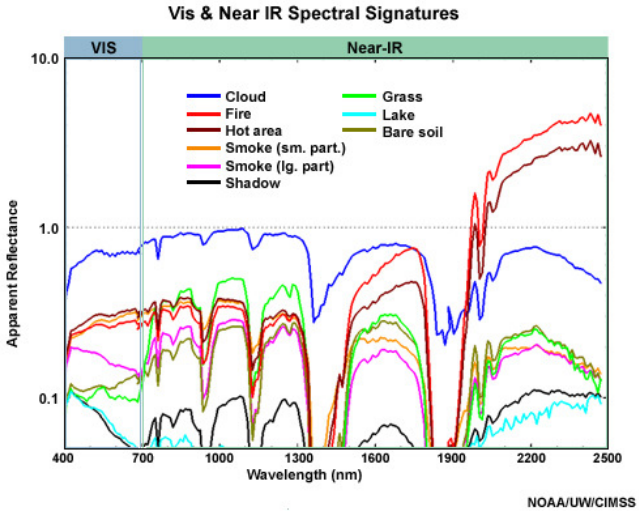


Figure 10.13.: Spectral signatures of different compounds

As seen on chapter 8, the pixels on a sensor are sensitive to a relatively wide part of the light spectra. In order to simulate the human perception, a mosaic composed by three different types of filters is placed in front of the sensor, generating a 3 channel RGB image.

The trichromatic paradigm falls short in detecting components with narrow spectral signatures, such as some chemicals or even bacterias [LPS+10, FVG01]. Figure 10.13 from [Kö] shows the spectral signature of different components: for such applications tens or even hundreds of narrow colors are needed.

Qtec Qsample (Figure 10.14) is a research platform for acquiring and analysing hyperspectral images. It is designed for scientific applications and for exploring the solution space of industrial applications. It is based on a QT5022 platform with the following configuration:

- Light: Halogen.
- Lens/Optics: C-Mount Fixed Lens.
- Imaging Sensor: 1 x 2Mpix CMOSIS IMEC Snapshot sensor.

Figure 10.14.: Qtec Qsample

- Processing Unit: AMD G-T65N APU.

- Acceleration Unit: AMD Radeon E8860.

The system is composed by a camera mounted over a conveyor belt with an encoder controlled by the QT5022 with Ethercat.

The system is designed with maximum customization in mind: There are 3 different belt colors (white, blue and black) interchangeable by the user to increase the contrast of the product. The speed and acceleration of the conveyor belt motor can be changed on the fly. Two extra cameras can be mounted over the platform and even the head of each camera can be replaced to test different sensors.

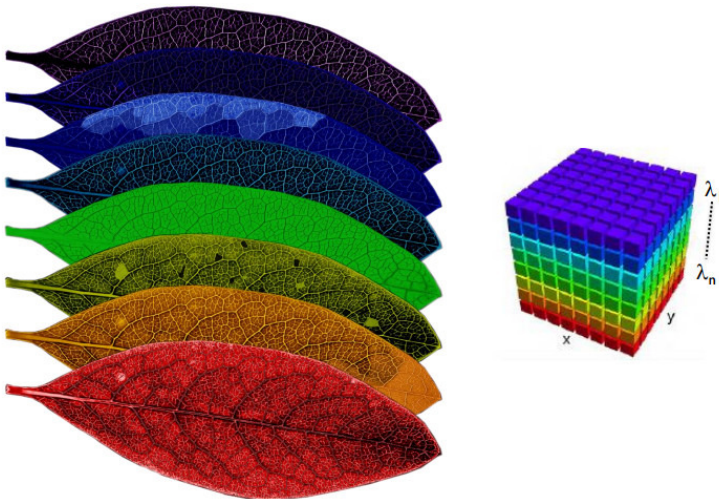


Figure 10.15.: Hyperspectral cube

10.4.1 *Hyperspectral cube*

Hyperspectral images are usually represented as data cubes, like shown in the Figure 10.15. Each layer of the cube corresponds to one of the "colors", usually much narrower than the trichromatic sensors.

Typical hyperspectral cameras make use of a spectrograph on the optical path. [DFRR+10, ADBLL09, ADBL11]. Such configuration has the problem of only inspecting one line of the scene. In order to get the data cube, multiple frames need to be captured and then stitched.

The sensor selected for this application, has a filter mosaic with a configuration of 5x5 pass-band filters. With this sensor a 25 channel cube is captured with a single image.

However, acquiring the data cube is only half of the problem: the interpretation of the data cube is not intuitive; specific visualization and processing tools are required.

Among the QT5022 software stack, there is a port of Gerbil, an Open Source hyperspectral visualization tool, from the Pattern Recognition Lab of the University of Erlangen-Nuremberg [JA10]. A screenshot of

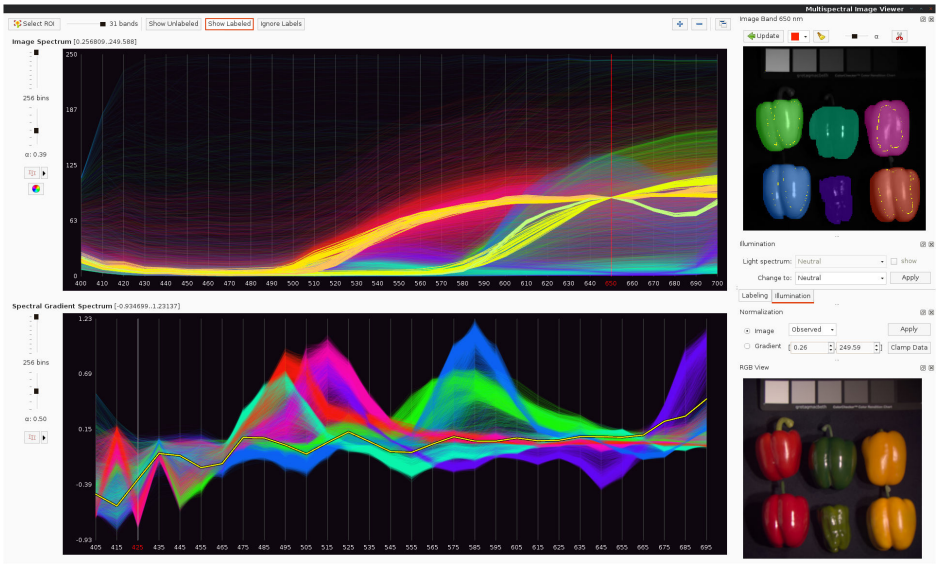


Figure 10.16.: Gerbil

the tool can be observed on figure 10.16. Gerbil makes use of a custom file format, consisting on one image per band and a text file describing what is the spectral range of each image. In this application, a custom GStreamer sink plugin has been developed for generating this file format.

Aside from Gerbil, Python (Figure 9.5) and OpenCV are also available as part of QT5022 stack.

10.4.2 *Results*

Qsample has been used during the requirement analysis of multiple Computer Vision projects, such as, screw counting, early detection of chocolate defects and bacteria analysis on raw materials.

In all of those projects, the scientist/engineer has been able to identify the key elements of the project and give valid estimations.

Other great advantage of this product is that it has all the elements required for an easy transition from the prototype to the final product: The user can experiment with different sensors and light sources, and even start the software development before the mechanical part of the final product is completed.

Part IV

CONCLUSIONS

11

CONCLUSIONS AND FUTURE WORK

Contents

11.1 CONTRIBUTIONS TO COMPUTER VISION SYSTEMS

This Thesis introduces a Novel Computer Vision System based on a modular structure. This system has been validated with four different real life applications that are currently deployed in hundreds of sites around the world.

The main three contributions of this thesis are the following:

- Definition of a modular Computer Vision System.
- Xform, an FPGA synthesizable image processing algorithm.
- Contributions to the Open Source Community.

11.1.1 *Modular Computer Vision System*

This Thesis identifies the key components of a Computer Vision System: Sensor, Image Processing Pipeline, Processing Unit, Acceleration Unit and Computer Vision Stack.

The document describes a series of interfaces, based on Industry Standards that can be used to interconnect those components.

This modular approach allows component interchange, reutilization of elements and result comparison in a fair way.

11.1.2 *Xform*

This Thesis describes a highly configurable computer vision algorithm that can perform geometrical and illumination transformations on the fly.

Among the multiple operations that the algorithm can do are: perspective correction, rotation, camera calibration, illumination compensation, resizing and cropping.

The algorithm is designed for its implementation on FPGAs with limited resources. On a Xilinx Spartan 6 it is capable of running with a pixel rate of 165 MHz, and a latency of 90 horizontal lines.

11.1.3 *Open Source*

The common components of the open source stack were not originally designed for Industrial Computer Vision Applications.

The author of this Thesis has worked with the Open Source community to implement new interfaces and functionalities that can fulfill the requirements of the Computer Vision Industry.

As a result of this work, more than 200 changes have been accepted on different Open Source Projects like the Linux Kernel, Yocto Project and U-boot, among others.

These contributions are already used by other systems.

11.2 SYSTEM IMPLEMENTATION: QT5022

The modular system described on this document has been implemented on a real life product: The Qtec QT5022. Today, this product constitutes a mature ecosystem with 19 different sensors, 2 Image Processing Units, 2 Acceleration Units and a Software Stack composed by over 11000 packages.

Its ease of use and configurability has facilitated the creation new prototypes and products. QT5022 is present in more than 7 industrial applications and it is deployed in over 200 sites around the world: Denmark, Sweden, United Kingdom, Germany, Netherlands, France, USA, Canada, Israel and Australia.

One the users of the platform, Newtec Engineering, have reported that thanks to this platform in one year they have been able to develop more new applications on the last three years combined.

11.3 CONTRIBUTIONS

11.3.1 *Publications*

Most of the results and concepts in this Thesis have been published in Scientific Magazines, Specialized Press or in Conferences. This is a list of those contributions and their relation with the document.

Embedded System on Chip Computer Vision:

- Ribalda, R., De Rivera, G. G., De Castro, Á., & Garrido, J. (2010). A mobile biometric system-on-token system for signing digital transactions. IEEE Security & Privacy, (2), 13-19.
- Morales, A., Ferrer, M. Á., Faundez, M., Fàbregas, J., Gonzalez, G., Garrido, J., Ribalda. R. & Freire, M. (2009). Biometric system verification close to "real world" conditions. In Biometric ID Management and Multimodal Communication (pp. 236-243). Springer Berlin Heidelberg.

FPGA Computer Vision System:

- Ribalda, R. (2007). Intelligent and reconfigurable architecture for remote image processing applications based on FPGAs and Linux. Advance Studies Thesis, UAM
- Ribalda, R., De Castro, A., Glez-de-Rivera, G., & Garrido, J. (2008, March). Open and Reconfigurable System on Chip Architecture with Hardware and Software Preprocessing Capabilities Used for Remote Image Acquisition. In Programmable Logic, 2008 4th Southern Conference on (pp. 167-172). IEEE.
- Fierrez, J., Galbally, J., Ortega-Garcia, J., Freire, M. R., Alonso-Fernandez, F., Ramos, D., Ribalda. R, ... & Gracia-Roche, J. J. (2010). BiosecurID: a multimodal biometric database. Pattern Analysis and Applications, 13(2), 235-246.

GPU Computer Vision System:

- Kleinert, A., Friedl-Vallon, F., Guggenmoser, T., Höpfner, M., Neubert, T., Ribalda, R., ... & Preusse, P. (2014). Level 0 to 1 processing of the imaging Fourier transform spectrometer GLORIA: generation of radiometrically and spectrally calibrated spectra. Atmospheric measurement techniques, 7(12), 4167-4184.

Modular Computer Vision System:

- de Rivera, G. G., Ribalda, R., Colás, J., & Garrido, J. (2005, July). A generic software platform for controlling collaborative robotic system using XML-RPC. In Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on (pp. 1336-1341). IEEE. Chicago.

- Madsen, K. & Ribalda.R (2015, February). APU vs. FPGA Was setzt sich bei intelligenten Kameras durch?. inVision. TeDo Verlag Germany.

- Madsen, K. & Ribalda.R (2015, August) APUs vs FPGAs: The Battle for Smart Camera Processing Supremacy. Electronic Design, Penton Electronics Group, USA.

- Ribalda, R. (2013, November) New V4L2 API: Multiple selections. Linux Kernel Media Workshop, Kernel Summit, Linux Foundation. Edinburgh

- Ribalda, R. (2014, October) New V4L2 API Proposals: Multiple timestamps & Dead pixels. Linux Media Summit Summit, Linux Foundation. Düsseldorf

- Ribalda, R. (2015 October) The Art of Counting Potatoes with Linux. Embedded Linux Conference Europe. Linux Foundation. Dublin

- Ribalda, R. (2015 October) Poster: Modular Industrial Camera with Linux. Embedded Linux Conference Europe. Linux Foundation. Dublin

11.3.2 *Open Source*

This work constitutes a great commitment to the Open Source Software.

In Computer Science, papers and conferences are a good instrument to introduce innovation, but they only reach a small amount of the population.

On the other hand, Open Source projects are used on daily basis by millions of devices and users and can also be used as a vehicle for sharing scientific results: The Linux Kernel have implementations of experimental file systems, schedulers, network protocols; OpenCV is the preferred tool for releasing research algorithm....

During the development of this Thesis, any change or component that could be reused, has been shared with the Open Source Community and, after the peer-review process, merged into their codebase.

The changes to the Kernel introduced by this Thesis are used by millions of android users around the world. Other companies like Intel, Samsung, Scientific Research Corporation, Xilinx are building new products using changes introduced by this Thesis.

The following projects have accepted changes from this Thesis:

- Linux Kernel: 172 contributions merged. Including a 9+ year old bug. 2nd Spanish Contributor by number of patches.
- U-boot: 25 contributions. Maintainer of Virtex PowerPC boards.
- Yocto project: 17 contributions. Supporting organization of the project.
- V4l-utils / libv4l2: 7 contributions.
- Flashrom: Support for the first board with EEprom memory.
- Clpeak: 2 contributions.
- Video Lan Client: 1 contribution.

All these changes are detailed on the Appendix of this document.

11.4 FUTURE WORK

So far, the Vision System presented in this Thesis has been successfully used in all the applications that it has been tested on. But in order to be ready for the requirements of the future applications it needs to be constantly updated. Thanks to the modular architecture of the platform, this can be done gradually. The following paragraphs outline the main improvements to the platform scheduled in the near future:

From the point of view of the user, the Image Processing Pipeline is a black box. This is mainly due to the complexity of the development of new cores in VHDL. The new generation of Xilinx Tools allows an easy development of cores with the use High Level Languages like C. It is planned to create an easy path for integrating user cores on the platform: this includes the tools for partial reconfiguring the FPGA and a framework for creating drivers for such cores easily.

AMD is currently working on a new generation of drivers where all the kernel code is Open Source and all the Proprietary parts of the stack are isolated in an optional userspace BLOB. This new driver, called AMDGPU is going through the review process for its acceptance in the next release of the kernel (4.3) and will only support the APU families from Carrizo (2015). This new generation of driver will be much more integrated with the rest of the applications on the stack, and will be easily debugged and optimized by the Open Source Community. In order to use this driver, a new Processing Unit should be developed with a newer AMD APU family. This project is on hold until AMD sells these chips in small-medium volume (less than 1.000 units).

During the development of this Thesis, 19 different sensors have been ported into the platform. Some of this sensors, like the Microbolometer, are very rare and have exotic requirements in terms of electronic and readout logic. Due to these requirements and their low production volume, it is very difficult to find commercial products based on them. On the near future it is planned to create a new Image Processing Pipeline with USB interface that will allow the use of these sensors by a standard PC, or even by another QT5022 camera as an external sensor.

Finally, it would be ideal that this platform would be broadly adopted by the Academia and the Industry, which would develop new modules independently. In this aspect, the platform is currently being presented to Danish research centers, which will hopefully use it to solve the Computer Vision problems of the future. Hopefully, one day, experts from the different areas of computer vision: optics, vision, electronics, software, will finally have a common platform to share their achievements and results.

12

CONCLUSIONES Y LÍNEAS FUTURAS

Contents

12.1 CONTRIBUCIONES A LOS SISTEMAS DE VISIÓN POR COMPUTADOR

Esta tesis presenta un original sistema de visión por computador basado en una estructura modular. Dicho sistema ha sido validado en cuatro aplicaciones industriales que son usadas en cientos de instalaciones alrededor del mundo.

Las tres principales contribuciones de esta tesis son las siguientes:

- Definición de un sistema modular de Visión por Computador.
- Xform, un algoritmo de procesado de imagen sintetizable en FPGA.
- Diferentes contribuciones a la comunidad de software libre.

12.1.1 *Sistema modular de visión por Computador*

La presente tesis identifica los elementos principales de un sistema de visión por computador: Sensor, procesador de imágenes "al vuelo", unidad de procesado principal, acelerador hardware y pila de software.

A lo largo del documento se describen las interfaces entre los mismos, todas ellas basadas en estándares industriales.

Este diseño modular permite el intercambio y la reutilización de componentes, así como comparar resultados de forma objetiva.

12.1.2 *Xform*

Esta tesis describe un algoritmo de visión por computador para realizar transformaciones geométricas y de iluminación "al vuelo", con un alto nivel de configurabilidad.

Entre las múltiples operaciones que el algoritmo puede desarrollar están: Corrección de perspectiva, calibración de cámara, compensación de iluminación, cambio de tamaño y recorte.

Este algoritmo está diseñado para su implementación en FPGAs de recursos limitados. En una Xilinx Spartan 6 puede funcionar con una tasa de píxeles de 165 MHz y una latencia de 90 líneas horizontales.

12.1.3 *Software Libre*

Las principales aplicaciones que componen la pila de Software Libre estándar no fueron originalmente desarrolladas para cumplir los requerimientos de las aplicaciones de visión por computador industrial.

El autor de esta tesis ha trabajado junto a la comunidad de Software libre para implementar las interfaces y funcionalidades requeridas para su aplicación en los entornos industrial.

Como resultado de este trabajo, más de 200 cambios han sido aceptados en algunos de los proyectos de software libre más representativos, tales como, el Kernel de Linux, Yocto Project y U-boot.

Estas contribuciones ya están siendo usadas por otros sistemas de visión por computador.

12.2 IMPLEMENTACIÓN DE LA PLATAFORMA : QT5022

El sistema modular descrito en este documento ha sido implementado en un producto: Qtec Qt5022. A día de hoy, esta plataforma constituye un ecosistema maduro con soporte para más de 19 sensores distintos, 2 procesadores de imágenes al vuelo y una colección de software con más de 11.000 paquetes.

Su facilidad de uso y gran configurabilidad, han permitido la creación de nuevos prototipos y productos basados en esta plataforma. Actualmente, QT5022 es usado en más de 7 productos industriales distintos, emplazados en más de 200 instalaciones alrededor del mundo en los siguientes países: Dinamarca, Suecia, Reino Unido, Alemania, Holanda, Francia, Estados Unidos, Canadá, Israel y Australia.

Newtec Engineering, uno de los usuarios de la plataforma, ha manifestado que gracias a esta plataforma, durante este año han podido desarrollar más productos distintos que en los últimos 3 años juntos.

12.3 CONTRIBUCIONES

12.3.1 *Publicaciones*

La gran mayoría de los resultados y conceptos presentados en este documento han sido publicados en revistas científicas, prensa especializada o divulgados en conferencias. A continuación se muestra la lista de éstas publicaciones junto con su relación con el documento.

Sistema de Visión por Computador basado en una plataforma System on Chip:

- Ribalda, R., De Rivera, G. G., De Castro, Á., & Garrido, J. (2010). A mobile biometric system-on-token system for signing digital transactions. IEEE Security & Privacy, (2), 13-19.
- Morales, A., Ferrer, M. Á., Faundez, M., Fàbregas, J., Gonzalez, G., Garrido, J., Ribalda. R. & Freire, M. (2009). Biometric system verification close to "real world" conditions. In Biometric ID Management and Multimodal Communication (pp. 236-243). Springer Berlin Heidelberg.

Sistema de Visión por Computador basado en FPGA:

- Ribalda, R. (2007). Intelligent and reconfigurable architecture for remote image processing applications based on FPGAs and Linux. Advance Studies Thesis, UAM
- Ribalda, R., De Castro, A., Glez-de-Rivera, G., & Garrido, J. (2008, March). Open and Reconfigurable System on Chip Architecture with Hardware and Software Preprocessing Capabilities Used for Remote Image Acquisition. In Programmable Logic, 2008 4th Southern Conference on (pp. 167-172). IEEE.
- Fierrez, J., Galbally, J., Ortega-Garcia, J., Freire, M. R., Alonso-Fernandez, F., Ramos, D., Ribalda. R, ... & Gracia-Roche, J. J. (2010). BiosecurID: a multimodal biometric database. Pattern Analysis and Applications, 13(2), 235-246.

Sistema de Visión por Computador basado en GPU:

- Kleinert, A., Friedl-Vallon, F., Guggenmoser, T., Höpfner, M., Neubert, T., Ribalda, R., ... & Preusse, P. (2014). Level 0 to 1 processing of the imaging Fourier transform spectrometer GLORIA: generation of radiometrically and spectrally calibrated spectra. Atmospheric measurement techniques, 7(12), 4167-4184.

Sistema de Visión por Computador Modular

- de Rivera, G. G., Ribalda, R., Colás, J., & Garrido, J. (2005, July). A generic software platform for controlling collaborative robotic system using XML-RPC. In Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on (pp. 1336-1341). IEEE. Chicago.
- Madsen, K. & Ribalda.R (2015, February). APU vs. FPGA Was setzt sich bei intelligenten Kameras durch?. inVision. TeDo Verlag Germany.
- Madsen, K. & Ribalda.R (2015, August) APUs vs FPGAs: The Battle for Smart Camera Processing Supremacy. Electronic Design, Penton Electronics Group, USA.
- Ribalda, R. (2013, November) New V4L2 API: Multiple selections. Linux Kernel Media Workshop, Kernel Summit, Linux Foundation. Edinburgh
- Ribalda, R. (2014, October) New V4L2 API Proposals: Multiple timestamps & Dead pixels. Linux Media Summit Summit, Linux Foundation. Düsseldorf
- Ribalda, R. (2015 October) The Art of Counting Potatoes with Linux. Embedded Linux Conference Europe. Linux Foundation. Dublin
- Ribalda, R. (2015 October) Poster: Modular Industrial Camera with Linux. Embedded Linux Conference Europe. Linux Foundation. Dublin

12.3.2 *Software Libre*

Esta tesis representa una clara apuesta por el Software Libre.

En Ciencias de la Computación, como en cualquier otro ámbito científico, los papers y las conferencias son una gran herramienta para avanzar y compartir resultados, sin embargo únicamente son accesibles por una pequeña proporción de la población.

Por el contrario, el Software Libre es utilizado por millones de dispositivos y usuarios cada día, al mismo tiempo que puede ser usado como una vehículo para compartir resultados científicos, por ejemplo el Kernel de Linux tiene implementaciones de sistemas de fichero, planificadores y protocolos de red experimentales. Asimismo, OpenCV es una de las aplicaciones más usadas para distribuir demostraciones de algoritmos de visión por computador experimentales.

Durante el desarrollo de esta tesis, cualquier pieza de software que fuera susceptible de ser interesante para cualquier otra persona, se ha compartido con la comunidad de Software Libre, y, tras un proceso de revisión de pares, incorporado a los diferentes proyectos que la componen.

Los cambios introducidos en el Kernel de Linux, como resultado de esta tesis, ya son usados por millones usuarios de Android en todo el mundo. Asimismo, otras compañías como Intel, Samsung Scienctific Research Corporation o Xilinx están usando igualmente las aportaciones de ésta tesis para crear nuevos productos.

La siguiente lista representa los cambios que ya han sido aceptados por los distintos proyectos de Software Libre:

- *Linux Kernel:* 172 contribuciones aceptadas (incluyendo un parche para un error con más de 9 años en el core). Segundo contribuidor en España por número de parches.
- *U-boot:* 25 contribuciones aceptadas. Mantenedor de las placas Virtex PowerPC.
- *Yocto project:* 17 contribuciones aceptadas. Nominado como "Supporting Organization".

- *libv4l2/v4l-utils:* 7 contribuciones aceptadas.
- *flashrom:* Soporte para la primera placa con memoria tipo EEPROM.
- *clpeak:* 2 contribuciones aceptadas.
- *vlc:* 1 contribución aceptada.

Todos estos cambios se encuentran detallados en el apéndice de este documento.

12.4 LÍNEAS FUTURAS

El objetivo principal de esta tesis ha sido superado de forma satisfactoria puesto que, hasta el momento, todas las aplicación probadas han podido ser ejecutadas satisfactoriamente en la implementación propuesta. Pero inevitablemente, para poder soportar los requerimientos de las aplicaciones futuras, algunos de los elementos de esta arquitectura han de mejorarse. Por suerte, la estructura modular de la misma permite una actualización escalonada a la vez que se mantiene la compatibilidad con las aplicaciones pasadas.

En los siguientes párrafos se presentan los principales cambios proyectados para la plataforma:

A día de hoy, el procesador de imágenes "al vuelo" se comporta como una caja negra a los ojos del usuario, esto es debido principalmente al complejo desarrollo de cores con VHDL. La nueva familia de herramientas de Xilinx, permite la creación de cores con lenguajes de alto nivel como C, lo que simplifica en gran medida la creación de cores. Debido a esto, se está trabajando en la creación de una infraestructura para poder incluir cores del usuario en tiempo de ejecución.

AMD, está trabajando en una nueva generación de drivers en los que todo el código ejecutado es software libre, a excepción de un pequeño conjunto de librerías opciones, contenidas en un BLOB en espacio de usuario. Este nuevo diseño, denominado AMDGPU, está pasando el proceso de revisión para ser parte de la siguiente versión del Kernel

(4.3) y soportará únicamente las familias de GPUs, desde Carrizo (2015). Esta nueva generación de drivers es mucho más simple de depurar y gracias al trabajo colaborativo de la comunidad, dará mejores resultados que el bloque propietario actual. Para poder usar esta nueva generación de drivers, se ha de desarrollar una nueva unidad de procesado principal par. Por el momento, esto está supeditado a que AMD comience a distribuir chips de las citadas familias en cantidades inferiores a 1000 unidades.

Durante el desarrollo de esta tesis, 19 sensores diferentes han sido portados a la plataforma. Algunos de los mismos, como el microbolómetro, son muy exóticos y difíciles de encontrar en sistemas comerciales. El desarrollo de un sistema de procesado al vuelo, con conexión USB, permitiría el uso de dichos sensores con un PC estándar o incluso como un sensor externo para otro sistema QT5022. En la actualidad se está estudiando las distintas plataformas disponibles para construir dicho sistema de procesado al vuelo.

Por último, el mejor de los escenarios sería en el que esta plataforma sea adoptada por la industria y la academia, lo cual desembocaría en la creación de módulos por terceras partes. En la actualidad, la plataforma QT5022 está siendo presentada a distintos centros de investigación de Dinamarca, que puede que la usen para resolver los problemas de visión de computador del futuro.

Part V

APPENDICES

A

CONTRIBUTIONS TO THE LINUX KERNEL

As a direct result of this Thesis, more than 172 contributions have been accepted into the main kernel tree after an intense intense review process. Chapters 9 describes in detail the relationship of these contributions with this Thesis.

This chapter shows the list of accepted contributions up to 1st of September of 2015. The current list can be found on the Kernel Git Repository Browser.

According to KPS [kps15], these contributions place the author of this Thesis as the second most active Spanish contributor in the Linux Kernel, and in the top 5% globally.

As a result of this contributions, the author of this Thesis attended the Kernel Summit 2013 [Rib13] (invitation only conference), as well as the Media summit 2014 [Rib14].

Commits "PCI: Generate uppercase hex for modalias interface class" and "PCI: Generate uppercase hex for modalias var in uevent" are particularly interesting as they fix a 9+ year bug in the kernel core infrastructure, leading to this comment by Greg Kroah-Hartman, one of the main Kernel maintainers:

Oh wait, I see what you are worried about now, the mis-match for just the upper bits of the class value.

Yeah, that's a bug, sorry about that, a 9+ year old one, nice catch :)

```
commit 9eac2d4d5312d2ea05c0dbba8051b868fe9961a4
Author: Ricardo <ricardo.ribalda@gmail.com>
```

```
Date:   Tue Oct 18 21:35:25 2011 +0000

 ll_temac: Add support for ethtool

 This patch enables the ethtool interface. The implementation is done
 using the libphy helper functions.

 Signed-off-by: David S. Miller <davem@davemloft.net>

commit c752d04066a36ae30b29795f3fa3f536292c1f8c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Oct 20 09:43:26 2011 +0200

 perf symbols: Increase symbol KSYM_NAME_LEN size

 Fglrx propietary driver has symbol names over 128 chars (:S). This
 breaks the function kallsyms__parse.

 This fix increases the size of KSYM_NAME_LEN, so kallsyms__parse can
 work on such kernels.

 The only counterparty, is that such function requires 128 more bytes to
 work.

 Acked-by: Pekka Enberg <penberg@kernel.org>
 Cc: Anton Blanchard <anton@samba.org>
 Cc: David Ahern <daahern@cisco.com>
 Cc: Ingo Molnar <mingo@elte.hu>
 Cc: Paul Mackerras <paulus@samba.org>
 Cc: Pekka Enberg <penberg@kernel.org>
 Cc: Peter Zijlstra <a.p.zijlstra@chello.nl>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

commit 8d8bdfe8034399357df58b5f3e4da638a9e9a257
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon Nov 7 23:47:45 2011 +0000

 ll_temac: Add support for phy_mii_ioctl
```

```
 This patch enables the ioctl support for the driver. So userspace
 programs like mii-tool can work.

 Resend in merge window

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit 50ec1538fac0e39078d45ca5f8a5186621830058
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon Nov 7 23:31:58 2011 +0000

 net/temac: FIX segfault when process old irqs

 Do not enable the irq until the scatter gather registers are ready to
 handle the data. Otherwise an irq from a packet send/received before
 last close can lead to an access to an invalid memory region on the irq
 handler.

 Also, stop the dma engine on close.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit f79d7e6f6ae397caf219cef1392ca39b3ff10833
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon Nov 7 23:39:57 2011 +0000

 net/ll_temac: FIX : Wait for indirect wait to end

 While tracing down a connectivity problem on the temac I connected a
 probe to the Cross bar irq, and it was triggered when doing
 ifdown->ifup.

 This is fixed once waiting for the indirect write to end. Since it is
 not on the hot path there is no performance loss.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit 1056e4388b0454917a512618c8416a98628fc9ce
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue Aug 13 11:04:06 2013 -0300

 [media] v4l2-dev: Fix race condition on __video_register_device

 When 2 devices are registered at the same time, in Part 2 both will get
 the same minor number, making Part6 crash, because it cannot create a
 create a device with a duplicated minor:

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Sakari Ailus <sakari.ailus@iki.fi>
 Acked-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit af67384f011e81ea86aef8aec51e62e775432ea8
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 14 14:23:47 2013 -0700

 leds-pca9633: Add support for PCA9634

 Add support for PCA9634 chip, which belongs to the same family as the
 9633 but with support for 8 outputs instead of 4.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit a5cd98b796b6891edc324415034a7f89531c754f
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 14 14:23:48 2013 -0700

 leds-pca9633: Unique naming of the LEDs

 If there is more than one pca963x chips on the system and there are
 some LEDs without platform_data names, the driver wont be able to
 provide unique naming to them.
```

```
 This will cause led_class_dev_register to fail, unregistering all the
 LEDs of the chip.

 This patch adds the i2c address to the name of the unnamed LEDs, making
 them unique.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit a7d0e9884fd7594d4de5066add5135ac6bb55bd4
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 14 14:23:49 2013 -0700

 leds-pca9633: Add mutex to the ledout register

 To update an LED a register has to be read, updated and writen. If
 another LED whas been updated at the same time, this could lead into
 wrong updates.

 This patch adds a common mutex to all the leds of the same chip to
 protect the ledout register.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit 56a1740c21e4396164265c3ec80e29990ddcdc36
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 14 14:23:50 2013 -0700

 leds-pca9633: Rename to leds-pca963x

 The driver now supports the chips pca9633 and pca9634, therefore we
 rename the files to more generic and meaningul names

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit 8a6acd648c1dfdc82f1bbcd15f7777962054c268
```

```
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 14 14:23:51 2013 -0700

 leds-pca963x: Fix device tree parsing

 A malformed device tree could lead into a segmentation fault if the reg
 value of a led is bigger than the number of leds.

 A valid device tree could have only information about the last led of the
 chip. Fix the device tree parsing to handle those cases.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit 819585bc4811bc54e316dfe521ce163816fa0ad1
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Aug 28 04:39:29 2013 -0300

 [media] videobuf2: Fix vb2_write prototype

 struct v4_file_operations defines the data param as
 const char __user *data but on vb2 is defined as
 char __user *data.
 This patch fixes the warnings produced by this. ie:
 drivers/qtec/qtec_xform.c:817:2: warning: initialization from
 incompatible pointer type [enabled by default]
 drivers/qtec/qtec_xform.c:817:2: warning: (near initialization for
                 'qtec_xform_v4l_fops.write') [enabled by default]

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Sylwester Nawrocki <s.nawrocki@samsung.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit df23728118cd0f53070769e2ac26a255f66daa57
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Aug 2 10:19:59 2013 -0300

 [media] videobuf2-dma-sg: Allocate pages as contiguous as possible
```

```
 Most DMA engines have limitations regarding the number of DMA segments
 (sg-buffers) that they can handle. Videobuffers can easily spread
 through hundreds of pages.
 In the previous aproach, the pages were allocated individually, this
 could led to the creation houndreds of dma segments (sg-buffers) that
 could not be handled by some DMA engines.
 This patch tries to minimize the number of DMA segments by using
 alloc_pages. In the worst case it will behave as before, but most
 of the times it will reduce the number of dma segments

 Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Reviewed-by: Andre Heider <a.heider@gmail.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Sylwester Nawrocki <s.nawrocki@samsung.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 223012475968fb8dac866bff5b278e9311a36894
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Aug 2 10:20:00 2013 -0300

 [media] videobuf2-dma-sg: Replace vb2_dma_sg_desc with sg_table

 Replace the private struct vb2_dma_sg_desc with the struct sg_table so
 we can benefit from all the helping functions in lib/scatterlist.c for
 things like allocating the sg or compacting the descriptor.
 marvel-ccic and solo6x10 drivers, that use this API have been updated.

 Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Reviewed-by: Andre Heider <a.heider@gmail.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 [s.nawrocki@samsung.com: minor corrections of the changelog]
 Signed-off-by: Sylwester Nawrocki <s.nawrocki@samsung.com>

 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 7167cf0e8cd10287b7912b9ffcccd9616f382922
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue Oct 1 08:17:10 2013 +0200
```

```
 ll_temac: Reset dma descriptors indexes on ndo_open

 The dma descriptors indexes are only initialized on the probe function.

 If a packet is on the buffer when temac_stop is called, the dma
 descriptors indexes can be left on a incorrect state where no other
 package can be sent.

 So an interface could be left in an usable state after ifdow/ifup.

 This patch makes sure that the descriptors indexes are in a proper
 status when the device is open.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit 9cd00941f8c274e6ca03ed238d96ddb0be474b86
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Sep 18 15:56:14 2013 +0200

 perf symbols: Support for Openembedded/Yocto -dbg packages

 On OpenEmbedded the symbol files are located under a .debug folder on
 the same folder as the binary file.

 This patch adds support for such files.

 Without this patch on perf top you can see:

 no symbols found in /usr/lib/gstreamer-1.0/libtheoraenc.so.1.1.2,
 maybe install a debug package?

 84.56%  libtheoraenc.so.1.1.2       [.] 0x000000000000b346

 With this patch symbols are shown:

 19.06%  libtheoraenc.so.1.1.2       [.] oc_int_frag_satd_thresh_mmxext
 9.76%   libtheoraenc.so.1.1.2       [.] oc_analyze_mb_mode_luma
```

```
 5.58%   libtheoraenc.so.1.1.2       [.] oc_qii_state_advance
 4.84%   libtheoraenc.so.1.1.2       [.] oc_enc_tokenize_ac
 ...

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Ingo Molnar <mingo@kernel.org>
 Cc: David Ahern <dsahern@gmail.com>
 Cc: Ingo Molnar <mingo@redhat.com>
 Cc: Jiri Olsa <jolsa@redhat.com>
 Cc: Namhyung Kim <namhyung@kernel.org>
 Cc: Paul Mackerras <paulus@samba.org>
 Cc: Peter Zijlstra <peterz@infradead.org>
 Cc: Stephane Eranian <eranian@google.com>
 Cc: Waiman Long <Waiman.Long@hp.com>
 Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

commit 98c24dcd6392784b1cd0dca462abe6f91f0cadc9
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 05:39:35 2013 -0300

 [media] em28xx-video: Swap release order to avoid lock nesting

 vb2_fop_release might take the video queue mutex lock.
 In order to avoid nesting mutexes the private mutex is taken after the
 fop_release has finished.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit a3d94dafbba04f498beacec68a32b063bbf62d08
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 11:40:18 2013 -0300

 [media] ths7303: Declare as static a private function

 git grep shows that the function is only called from ths7303.c
 Fix this build warning:
 CC      drivers/media/i2c/ths7303.o
```

```
 drivers/media/i2c/ths7303.c:86:5: warning: no previous prototype for
 ‘ths7303_setval’ [-Wmissing-prototypes]

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
 Acked-by: Lad, Prabhakar <prabhakar.csengg@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit dc793175c5b532f343fe2224afd9189130da0004
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 11:21:30 2013 -0300

 [media] smiapp: Fix BUG_ON() on an impossible condition

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Sakari Ailus <sakari.ailus@iki.fi>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 202dfbdc2b88286463b9e05df4be1c0bce50af4a
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 15:48:38 2013 -0300

 [media] videobuf2-dma-sg: Fix typo on debug message

 num_pages_from_user and buf->num_pages were swapped.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit f956035ce70b8da9928aac1424485c3526fccb11
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Nov 8 07:08:45 2013 -0300

 [media] vb2: Return 0 when streamon and streamoff are already on/off

 According to the doc:
 If VIDIOC_STREAMON is called when streaming is already in progress,
```

```
 or if VIDIOC_STREAMOFF is called when streaming is already stopped,
 then the ioctl does nothing and 0 is returned.
 The current implementation was returning -EINVAL instead.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 1380f5754cb0cc4b765629b153fc0e7030b86da2
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon Nov 25 05:49:02 2013 -0300

 [media] videobuf2: Add missing lock held on vb2_fop_release

 vb2_fop_release does not hold the lock although it is modifying the
 queue->owner field.
 This could lead to race conditions on the vb2_perform_io function
 when multiple applications are accessing the video device via
 read/write API.

 Signed-off-by: Ricardo Ribalda <ricardo.ribalda@gmail.com>
 Acked-by: Hans Verkuil <hans.verkuil@cisco.com>
 Acked-by: Sylwester Nawrocki <s.nawrocki@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 50ac952d2263bd5d7812acf1839d57c34c6f8d9f
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue Nov 26 09:58:44 2013 -0300

 [media] videobuf2-dma-sg: Support io userptr operations on io memory

 Memory exported via remap_pfn_range cannot be remapped via
 get_user_pages.
 Other videobuf2 methods (like the dma-contig) supports io memory.
 This patch adds support for this kind of memory.
 v2: Comments by Marek Szyprowski
 -Use vb2_get_vma and vb2_put_vma
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit bdee6bdb67e84aaf26a163aced908f4ab6bbd06b
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 05:39:35 2013 -0300

 [media] em28xx-video: Swap release order to avoid lock nesting

 vb2_fop_release might take the video queue mutex lock.
 In order to avoid nesting mutexes the private mutex is taken after the
 fop_release has finished.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 06eb891edb4009245278a0ae50ccfd6fc99004d2
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Nov 6 11:40:18 2013 -0300

 [media] ths7303: Declare as static a private function

 git grep shows that the function is only called from ths7303.c
 Fix this build warning:
 CC      drivers/media/i2c/ths7303.o
 drivers/media/i2c/ths7303.c:86:5: warning: no previous prototype for
 ‘ths7303_setval’ [-Wmissing-prototypes]
int ths7303_setval(struct v4l2_subdev *sd, enum ths7303_filter_mode mode)
     ^

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Laurent Pinchart <laurent.pinchart@ideasonboard.com>
 Acked-by: Lad, Prabhakar <prabhakar.csengg@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>
```

```
commit f90580ca0133c533763a6cb3e632a21098a382df
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue Nov 26 05:31:42 2013 -0300

 [media] videodev2: Set vb2_rect's width and height as unsigned

 As discussed on the media summit 2013, there is no reason for the width
 and height to be signed.

 Therefore this patch is an attempt to convert those fields from __s32 to
 __u32.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Sakari Ailus <sakari.ailus@iki.fi> (documentation and smiapp)
 Acked-by: Lad, Prabhakar <prabhakar.csengg@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 548df7831adc34eca3ccefa29f768cef84315d6e
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Jan 8 05:01:33 2014 -0300

 [media] vb2: Check if there are buffers before streamon

 This patch adds a test preventing streamon() if there is no buffer
 ready.

 Without this patch, a user could call streamon() before
 preparing any buffer. This leads to a situation where if he calls
 close() before calling streamoff() the device is kept streaming.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reviewed-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 249f5a58bc844506fef2e9d5d55a88fbc708c5fa
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 8 05:01:33 2014 -0300
```

```
 [media] vb2: Check if there are buffers before streamon

 This patch adds a test preventing streamon() if there is no buffer
 ready.

 Without this patch, a user could call streamon() before
 preparing any buffer. This leads to a situation where if he calls
 close() before calling streamoff() the device is kept streaming.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit 84237bfb0b112a18a7c5d4d35a0663c97bdd14c6
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu May 15 14:28:45 2014 +0200

 usb: gadget: net2280: Fix NULL pointer dereference

 When DEBUG is enabled driver->driver.name is accessed, but driver
 can be NULL

 [  174.411689] BUG: unable to handle kernel NULL pointer dereference

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 227ae227c9352903d8bc4dc42e128da93aca4c79
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Apr 25 13:11:29 2014 -0300

 [media] videobuf2-dma-sg: Fix NULL pointer dereference BUG

 vb2_get_vma() copy the content of the vma to a new structure but set
 some of its pointers to NULL.

 One of this pointer is used by follow_pte() called by follow_pfn()
 on io memory.
```

```
 This can lead to a NULL pointer derreference.

 The version of vma that has not been cleared must be used.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <m.chehab@samsung.com>

commit c4128cac3557ddd5fa972cb6511c426cd94a7ccd
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu May 15 14:28:46 2014 +0200

 usb: gadget: net2280: Add support for PLX USB338X

 This patch adds support for the PLX USB3380 and USB3382.

 This driver is based on the driver from the manufacturer.

 Since USB338X is register compatible with NET2280, I thought that it
 would be better to include this hardware into net2280 driver.

 Manufacturer's driver only supported the USB33X, did not follow the
 Kernel Style and contain some trivial errors. This patch has tried to
 address this issues.

 This patch has only been tested on USB338x hardware, but the merge has
 been done trying to not affect the behaviour of NET2280.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Tested-by: Alan Stern <stern@rowland.harvard.edu>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit adc82f77bee3487651f8ad253fb1c8a7bf4ec658
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:03 2014 +0200

 usb: gadget: net2280: Add support for PLX USB338X
```

```
 This patch adds support for the PLX USB3380 and USB3382.

 This driver is based on the driver from the manufacturer.

 Since USB338X is register compatible with NET2280, I thought that it
 would be better to include this hardware into net2280 driver.

 Manufacturer's driver only supported the USB33X, did not follow the
 Kernel Style and contain some trivial errors. This patch has tried to
 address this issues.

 This patch has only been tested on USB338x hardware, but the merge has
 been done trying to not affect the behaviour of NET2280.

 Tested-by: Alan Stern <stern@rowland.harvard.edu>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit c2db8a8a01978a1ffad735f31268a1c9c81b413e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:04 2014 +0200

 usb: gadget: net2280: Dont use magic numbers

 Instead of using magic numbers use #defines

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 3e76fdcba6b2e30921d280704ea10764f150ec9c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:05 2014 +0200

 usb: gadget: net2280: Use BIT() macro

 Improves readability of the code

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 00d4db0e8539571382e9629c6bc5195263f6a960
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:06 2014 +0200

 usb: gadget: net2280: Use true/false instead of 1/0

 For bool variables

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 9a028e46fc0aeb0e5036ac1f4393653404242a1a
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:07 2014 +0200

 usb: gadget: net2280: Use module_pci_driver macro

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit a27f37a13cfbcfaeb987a910661a860f8d2f915e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:08 2014 +0200

 usb: gadget: net2280: Refactor queues_show

 Replace a long and ugly expresion with an already available function.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit fae3c158800339765a2580ac5d6236ae116ec5cb
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:09 2014 +0200

 usb: gadget: net2280: Pass checkpacth.pl test
```

```
 Fix Code Style using checkpatch.pl criteria

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit ae8e530a7e5d87592cb23996bee7fd6f1eb202ed
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:10 2014 +0200

 usb: gadget: net2280: Code Cleanup

 - Move logical continuations to end of line
 - Improve spacing

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit e56e69cc0ff4905914695f20c927aa71597be94c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:11 2014 +0200

 usb: gadget: net2280: Use pr_* function

 Driver was using custom functions WARNING, ERROR, DEBUG, instead of
  pr_err, pr_dgb...

 New ep_* macros have been created that use standard pr_* functions.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 2eeb0016c1242f275f9ebacc687ab639689f8bad
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 20 18:30:12 2014 +0200

 usb: gadget: net2280: Use quirks instead of pci id

 Use of quirks improve readability and will be easier to add new devices
 to this driver.
```

```
 Suggested-by: Alan Stern <stern@rowland.harvard.edu>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit b99b406c990def280d64ceb6739ac32001497a90
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Jul 4 11:27:03 2014 +0200

 usb: gadget: net2280: Fix typo on #ifdef

 Commit e56e69cc0ff4 ("usb: gadget: net2280: Use pr_* function")
 includes a editing mistake on one of the #ifdef.

 This patch fixes it.

 Reported-by: Paul Bolle <pebolle@tiscali.nl>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 61ab9efddf51cbc0d57356a4d650785cf5721fbe
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon Aug 4 11:11:49 2014 +0200

 net/usb/hso: Add support for Option GTM671WFS

 After this patch:

 [   32.985530] hso: drivers/net/usb/hso.c: Option Wireless
 [   33.000452] hso 2-1.4:1.7: Not our interface
 [   33.001849] usbcore: registered new interface driver hso

 root@qt5022:~# ls /dev/ttyHS*
 /dev/ttyHS0  /dev/ttyHS1  /dev/ttyHS2  /dev/ttyHS3  /dev/ttyHS4
 /dev/ttyHS5


 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Dan Williams <dcbw@redhat.com>
```

```
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit ac9d032e739f13ca04bfe6e0fd04bd114f72c6e2
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Aug 26 18:00:19 2014 +0200

 usb: gadget: net2280: Fix invalid handling of Reset irq

 Without this patch, some hosts keep restarting indefinitely the target.

 Fixes: ae8e530 (usb: gadget: net2280: Code Cleanup)
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit e676253b19b2d269cccf67fdb1592120a0cd0676
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Aug 5 11:45:59 2014 +0200

 serial/8250: Add support for RS485 IOCTLs

 This patch allow the users of the 8250 infrastructure to define a
 handler for RS485 configration.

 If no handler is defined the 8250 driver will work as usual.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 28e3fb6c4dce76d59a76755c4360d1cd5e0e226c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Jul 31 21:22:26 2014 +0200

 serial: Add support for Fintek F81216A LPC to 4 UART

 This patch lets you set the RS485 cappabilites of the device through
 TIOCSRS485 and TIOCGRS485 as defined on
 Documentation/serial/serial-rs485.txt
```

```
 In order to probe the device, the PNP id and the device id is used.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 0b4af1d94903143f88e541b00f028fa449a26f73
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 07:17:45 2014 +0200

 serial/8250_core: Add reference to uacess.h

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 [3/21] serial/8250: Add
 support for RS485 IOCTLs, adds a building error on arch m32r.

 All error/warnings:

    drivers/tty/serial/8250/8250_core.c: In function 'serial8250_ioctl':
 >> drivers/tty/serial/8250/8250_core.c:2859:3: error: implicit declaration
 of function 'copy_from_user' [-Werror=implicit-function-declaration]
       if (copy_from_user(&rs485_config, (void __user *)arg,
       ^
 >> drivers/tty/serial/8250/8250_core.c:2871:3: error: implicit declaration
 of function 'copy_to_user' [-Werror=implicit-function-declaration]
       if (copy_to_user((void __user *)arg, &up->rs485,
       ^
    cc1: some warnings being treated as errors

 Reported-by: kbuild test robot <fengguang.wu@intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a9a2eab5fb65232512adac58898eef835124a40e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 21:39:24 2014 +0200

 xtensa/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 [3/21] serial/8250: Add
 support for RS485 IOCTLs, adds support for RS485 ioctls for 825_core on
```

```
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Reported-by: kbuild test robot <fengguang.wu@intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 12b46b66f098de4b72ea6f14b8228d1e71ab9fd1
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 20:58:15 2014 +0200

 parisc/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 (serial/8250: Add
 support for RS485 IOCTLs), adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 8e63aee564229f95d1e1d7e5e21ffe2622f28f16
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 20:59:50 2014 +0200

 sh/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 (serial/8250: Add
 support for RS485 IOCTLs), adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 999156ada570cb4a2eaee42e47c9e659b5c577fb
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 20:37:59 2014 +0200
```

```
 sparc/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 (serial/8250: Add
 support for RS485 IOCTLs), adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: David S. Miller <davem@davemloft.net>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit f2589bff1ce8b94cebc044e5dfeac4d4e8701cbc
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Sep 9 21:39:24 2014 +0200

 xtensa/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 [3/21] serial/8250: Add
 support for RS485 IOCTLs, adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Reported-by: kbuild test robot <fengguang.wu@intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Chris Zankel <chris@zankel.net>

commit 1c84cd48a117486166f3597c081b170b76e5bd81
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Sep 10 10:57:08 2014 +0200

 mips/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 (serial/8250: Add
 support for RS485 IOCTLs), adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunaltely the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Reported-by: Markos Chandras <markos.chandras@imgtec.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 89ec3dcf17fd3fa009ecf8faaba36828dd6bc416
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 27 14:57:57 2014 +0200

 PCI: Generate uppercase hex for modalias interface class

 Some implementations of modprobe fail to load the driver for a PCI device
 automatically because the "interface" part of the modalias from the kernel
 is lowercase, and the modalias from file2alias is uppercase.

 The "interface" is the low-order byte of the Class Code, defined in PCI
 r3.0, Appendix D.  Most interface types defined in the spec do not use
 alpha characters, so they won't be affected.  For example, 00h, 01h, 10h,
 20h, etc. are unaffected.

 Print the "interface" byte of the Class Code in uppercase hex, as we
 already do for the Vendor ID, Device ID, Class, etc.

 [bhelgaas: changelog]
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
 Acked-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
 CC: stable@vger.kernel.org

commit 195311761e5492afdb2ab0454f51223bc75649dd
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Sep 19 09:18:13 2014 +0200

 asm/uapi: Add definition of TIOC[SG]RS485

 Commit: e676253b19b2d269cccf67fdb1592120a0cd0676 (serial/8250: Add
 support for RS485 IOCTLs), adds support for RS485 ioctls for 825_core on
 all the archs. Unfortunately the definition of TIOCSRS485 and
 TIOCGRS485 was missing on the ioctls.h file

 Reported-by: Guenter Roeck <linux@roeck-us.net>
 Reviewed-by: Guenter Roeck <linux@roeck-us.net>
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit aad31088dbfe6016147469974f0cd6c3f3201f3e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Oct 8 21:57:27 2014 +0200

 serial/max310x: Remove obsolete #ifset TIOC[SG]RS485

 Commit e676253b19b2 ("serial/8250: Add support for RS485 IOCTLs") added
 references to TIOC[SG]RS48 on 8250_core.c. This change triggered the
 need to define them in all the arches that uses tty/serial.

 This made #ifdef TIOC[SG]RS48 obsolete.

 Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Cc: linux-serial@vger.kernel.org
 Cc: linux-kernel@vger.kernel.org
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c015b4ad2ae57e5f07a5d4f746835356056c8e03
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Oct 8 21:57:28 2014 +0200

 serial/sc16is7xx: Remove obsolete #ifset TIOC[SG]RS485

 Commit e676253b19b2 ("serial/8250: Add support for RS485 IOCTLs") added
 references to TIOC[SG]RS48 on 8250_core.c. This change triggered the
 need to define them in all the arches that uses tty/serial.

 This made #ifdef TIOC[SG]RS48 obsolete.

 Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Cc: linux-serial@vger.kernel.org
 Cc: linux-kernel@vger.kernel.org
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a5f276f10ff70da89b349df445e944c8cd87956c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 22:46:13 2014 +0100

 serial_core: Handle TIOC[GS]RS485 ioctls.

 The following drivers: 8250_core, atmel_serial, max310x, mcf, omap-serial
 and sci16is7xx implement code to handle RS485 ioctls.

 In order to avoid code duplication, we implement a simple ioctl handler
 on the serial_core layer.

 This handler can be used by all the other drivers instead of duplicating
 code.

 Until this is the only RS485 ioctl handler, it will try first the
 rs485_config callback and if it is not present it will call the driver
 specific ioctl.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 46c55b4bb9b55b7b09b6879668a209a5657814b3
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:51 2014 +0100

 serial/8250: Copy RS485 fields to serial_core

 Initialize recently added rs485 fields on serial_core

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Cc: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
 Cc: Alan Cox <alan@linux.intel.com>
```

```
 Cc: Tony Lindgren <tony@atomide.com>
 Cc: Peter Hurley <peter@hurleysoftware.com>
 Cc: Yoshihiro YUNOMAE <yoshihiro.yunomae.ez@hitachi.com>
 Cc: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
 Cc: Ingo Molnar <mingo@elte.hu>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 41e69093fd608c0348fb12c0879ab464f986d2a8
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:52 2014 +0100

 8250/fintek: Use rs485 handler from serial_core

 In order to remove the handler for rs485 ioctls on serial_8250, all the
 drivers must use the implementation on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 039ec1f010e6b058f497381d5a6bb840e160b4ac
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:53 2014 +0100

 serial/8250: Remove obsolete handling of rs485 ioctls

 There is no more users for this functions. All the 8250 drivers are
 using the rs485 handler on serial_core instead.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Cc: Sebastian Andrzej Siewior <bigeasy@linutronix.de>
 Cc: Alan Cox <alan@linux.intel.com>
 Cc: Tony Lindgren <tony@atomide.com>
 Cc: Peter Hurley <peter@hurleysoftware.com>
 Cc: Yoshihiro YUNOMAE <yoshihiro.yunomae.ez@hitachi.com>
 Cc: Andy Shevchenko <andriy.shevchenko@linux.intel.com>
```

```
 Cc: Ingo Molnar <mingo@elte.hu>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b57d15fe8b37fef8f374afa08e3a557898d5d517
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:54 2014 +0100

 serial/sc16is7xx: Use the rs485 functions on serial_core

 In order to unify all the rs485 ioctl handling.
 Use the implementation of TIOC[GS]RS485 ioctl handling on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 2fc0184dae7be565e4ad47c720e6014cd5543a21
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:55 2014 +0100

 serial/mcf: Use the rs485 functions on serial_core

 In order to unify all the rs485 ioctl handling.
 Use the implementation of TIOC[GS]RS485 ioctl handling on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 13bd3e6fa177883914fa64b609651e56d58eea65
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:56 2014 +0100

 serial/atmel: Use the rs485 functions on serial_core

 In order to unify all the rs485 ioctl handling.
```

```
 Use the implementation of TIOC[GS]RS485 ioctl handling on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Nicolas Ferre <nicolas.ferre@atmel.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dadd7ecbff4bf01ec446c4390cfeab20124b5708
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 22:46:14 2014 +0100

 serial/omap: Use the rs485 functions on serial_core

 In order to unify all the rs485 ioctl handling
 Use the implementation of TIOC[GS]RS485 ioctl handling on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit c267d679cfd9699b9349fd714f63f6b4ee59dda2
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:22:58 2014 +0100

 drivers/max310: Use the rs485 functions on serial_core

 In order to unify all the rs485 ioctl handling
 Use the implementation of TIOC[GS]RS485 ioctl handling on serial_core.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a9c20a9cf3190a517b88d8e08d93157256f97673
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
Date:   Thu Nov 6 09:22:59 2014 +0100

 serial_core: Remove call to driver-specific TIO[GS]RS485]

 Once there is no more handlers for TIOC[GS]RS485 there is no need to
 call the driver specific ioctl when the generic implementation is
 missing.

 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit bd737f8738b7e15930aa7b47c94c28a8d83148ac
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Nov 6 09:23:00 2014 +0100

 tty/serial_core: Introduce lock mechanism for RS485

 Introduce an homogeneous lock system between setting and using the rs485
 data of the uart_port.

 This patch should not be split into multiple ones in order to avoid
 leaving the tree in an unstable state.

 Acked-by: Nicolas Ferre <nicolas.ferre@atmel.com>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Suggested-by: Alan Cox <alan@linux.intel.com>
 Cc: Nicolas Ferre <nicolas.ferre@atmel.com>
 Cc: Jiri Slaby <jslaby@suse.cz>
 Cc: One Thousand Gnomes <gnomes@lxorguk.ukuu.org.uk>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a831f20b6d6460640b83644d1c1df6e7e8ca9f68
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Nov 24 11:19:51 2014 +0100

 wireless/p54: Remove duplicated net2280 header
```

```
    The usb gadget driver net2280 has exported a header file with the
    register definition of the net2280 chip.

    Remove the custom/duplicated header file in favor of that header file
    in include/linux

    Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
    Signed-off-by: John W. Linville <linville@tuxdriver.com>

commit 145b3fe579db66fbe999a2bc3fd5b63dffe9636d
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Dec 2 17:35:04 2014 +0100

    PCI: Generate uppercase hex for modalias var in uevent

    Some implementations of modprobe fail to load the driver for a PCI device
    automatically because the "interface" part of the modalias from the kernel
    is lowercase, and the modalias from file2alias is uppercase.

    The "interface" is the low-order byte of the Class Code, defined in PCI
    r3.0, Appendix D.  Most interface types defined in the spec do not use
    alpha characters, so they won't be affected.  For example, 00h, 01h, 10h,
    20h, etc. are unaffected.

    Print the "interface" byte of the Class Code in uppercase hex, as we
    already do for the Vendor ID, Device ID, Class, etc.

    Commit 89ec3dcf17fd ("PCI: Generate uppercase hex for modalias interface
    class") fixed only half of the problem.  Some udev implementations rely on
    the uevent file and not the modalias file.

    Fixes: d1ded203adf1 ("PCI: add MODALIAS to hotplug event for pci devices")
    Fixes: 89ec3dcf17fd ("PCI: Generate uppercase hex for modalias
    interface class")
    Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
    Signed-off-by: Bjorn Helgaas <bhelgaas@google.com>
    Acked-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    CC: stable@vger.kernel.org
```

```
commit 906641980c3ae4e4175268168897ebdc7314c73c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:46 2014 +0100

 usb: gadget: udc: net2280: Remove obsolete module param use_dma_chaining

 use_dma_chaining module parameter was designed to avoid creating one irq
 per package on a group of packages (with the help of the driver's flag
 no_interrupt).

 Unfortunately, when this parameter is enabled, the driver fails to work
 on both net2280 and 3380 chips.

 This patch removes this parameter, which was disabled by default.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 7a74c48172d8fad6da0f9be5d353cf322305af07
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:47 2014 +0100

 usb: gadget: udc: net2280: remove full_speed module parameter

 This patch removes the full_speed parameter used force full-speed
 operation.

 It was designed exclusively for testing purposes, and there is no
 reason to maintain this in a production kernel.

 Reverts: 2f0760774711c957c395b31131b848043af98edf
 Suggested-by: Alan Stern <stern@rowland.harvard.edu>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 9c864c23424fa6a2e23b8e81faa6a100ab01481c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:48 2014 +0100
```

```
 usb: gadget: udc: net2280: Remove module parameter use_msi

 Parameter use_msi was used to enable msi irq on usb338x chips, it was
 enabled by default.

 There is no reason to prefer non-msi irq on usb338x, and it falls back
 to non msi on error.

 Therefore remove this option.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit d588ff58c20d8503f789b9cbbed55e796ae6d04e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:49 2014 +0100

 usb: gadget: udc: net2280: Remove use_dma module parameter

 use_dma parameter was designed to enable the dma on the chip. It was
 enabled by default.

 It comes from the time when the dma was not reliable. Now it has been
 working ok in production.

 This patch removes this parameter.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit cb52c698c6030c88733c87def5c359a09b76bc1e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:50 2014 +0100

 usb: gadget: udc: net2280: Remove dma_started field

 Remove dma_started field from net2280_ep structure, since it is not used
 by any function.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 61e72dc65a9a4937d351a4b67386a6bbf70a1146
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:51 2014 +0100

 usb: gadget: udc: net2280: Remove restart_dma inline function definition

 restart_dma is not used before it is declaration. Therefore we can
 remove this definition.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit cf8b1cdebab3c6fb01d37e62e14f9dd8cc99d09b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:52 2014 +0100

 usb: gadget: udc: net2280: Code cleanout remove ep_stdrsp function

 ep_stdrsp was only called by handle_stat0_irqs_superspeed and with
 always the same flags.

 Remove the function and replace the call by the code inside the
 function, since it is very simple once the dead code is removed.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 5d1b6840fd871b51765e7edee82876f760c46adf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:53 2014 +0100

 usb: gadget: udc: net2280: Remove field is_halt

 Field is_halt is never used by any function.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit e0cbb04627556a953c6fe8d5ba42b9dda68146af
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:54 2014 +0100

 usb: gadget: udc: net2280: Remove function ep_stall

 irqs_superspeed calls ep_stall instead of set/clear_halt, due to a
 workaround for SS seqnum. Create a function with the workaround and
 call set/clear_halt instead.

 This way we can compare the code of super/normal speed and it is easier
 to follow the code.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit e721c4575db77a543d0a28c56e95b84ee37a551c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:55 2014 +0100

 usb: gadget: udc: net2280: Merge abort_dma for 228x and 338x

 We can use the same function for both families of chips and also remove
 the ep_stop_dma() function.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 485f44d06b352a788d07e79a5744cc7420adc1e4
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:56 2014 +0100

 usb: gadget: udc: net2280: Clean function net2280_queue

 Do not duplicate the code for the else branch of the condition.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 5517525e05b5d21c26e2b729a7e977f7d69714af
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:57 2014 +0100

 usb: gadget: udc: net2280: Improve patching of defect 7374

 Once the defect 7374 is patched, there is no reason the keep reading the
 idx scratch register.

 Cache the content of the scratch idx register on device flag.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 5153c219e77985be2fb861814184729bb40b0987
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:58 2014 +0100

 usb: gadget: udc: net2280: Remove function resume_dma

 Function resume_dma is not used, remove it.

 The reason the compiler did not catch this dead code is the inline
 modifier.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 6897d4b2bafe189863b2fe448c4afde37844cdbf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:50:59 2014 +0100

 usb: gadget: udc: net2280: Declare allow_status_338x as inline

 The function is very simple, does not declare any variable and it is
 called in the irq path.
```

```
 The counterpart for net228x is already declared as inline.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit 43780aaa1c23c572ef57267d43d520affa7b4723
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:51:00 2014 +0100

 usb: gadget: udc: net2280: Simplify scan_dma_completions

 After fix superspeed dma_done was applied we can simplify the code by
 removing the duplicated dma_done and letting the function check if there
 are more completed dma transactions.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit d82f3db266f3be7f85de33905dba2a6caccb97f0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:51:01 2014 +0100

 usb: gadget: udc: net2280: Move ASSERT_OUT_NAKING into out_flush

 ASSERT_OUT_NAKING was only called by out_flush and was hidden behind a
 ifdef.

 This patch moves the inline function into out_flush and remove the
 ifdef. The user can decide to print the debug message or not via dynamic
 printk

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit cb442ee1592d26815156201afa986a4ec3fb4bbf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Nov 28 14:51:02 2014 +0100
```

```
 usb: gadget: udc: net2280: Re-enable dynamic debug messages

 Some debug messages were not build due to unconditional #if 0.

 These messages are very useful for debugging and the user can enable
 them on demand via dynamic debug.

 If they are not enabled the performance is not affected.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit fc12c68b4ff2904c3322b7a12089f0b9808e2383
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jan 13 10:54:58 2015 +0100

 usb: gadget: net2280: Dont use 0 as NULL pointer

 Fix sparse warning

 Fixes: cb442ee1592d2681 (usb: gadget: udc: net2280: Re-enable dynamic
 debug messages)
 Reported-by: kbuild test robot <fengguang.wu@intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Felipe Balbi <balbi@ti.com>

commit bc2f3dc3e2c784efc30acf6f8adde560beedbf57
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:08 2014 +0100

 gpio/xilinx: Remove offset property

 Instead of calculating the register offset per call, pre-calculate it on
 probe time.

 Acked-by: Alexandre Courbot <acourbot@nvidia.com>
 Acked-by: Michal Simek <michal.simek@xilinx.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
```

```
commit 749564ffd52d91ddf9917315e6fba2a3dcf3137e
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:09 2014 +0100

 gpio/xilinx: Convert the driver to platform device interface

 This way we do not need to transverse the device tree manually and we
 support hot plugged devices.

 Also Implement remove callback so the driver can be unloaded

 Acked-by: Michal Simek <michal.simek@xilinx.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit c54c58bad6e64649dfe51c2e8d9e5a1524d673e8
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:10 2014 +0100

 gpio/xilinx: Add support for X86 Arch

 Core can be accessed via PCIe on X86 platform.
 This patch also allows the driver to be used as module.

 Acked-by: Michal Simek <michal.simek@xilinx.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit 4ae798fae200f1d26c820c6e508a6bcd3c5aa8f2
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:11 2014 +0100

 gpio/xilinx: Fix kernel-doc

 Some documentation were not following the kernel-doc format.
 Backporting patch from Xilinx git repository.

 Suggested-by: Michal Simek <michal.simek@xilinx.com>
```

```
 Acked-by: Michal Simek <michal.simek@xilinx.com>
 Acked-by: Alexandre Courbot <acourbot@nvidia.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit 1d6902d3a68e83061c979a57d7411e8ae172df67
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:12 2014 +0100

 gpio/xilinx: Create a single gpio chip on dual cores

 Currently, we had two gpio chips on cores configured as dual.

 This lead to mapping the same memory region twice and duplicating the
 init and remove code.

 This patch creates a single gpiochip for single and dual cores.

 Suggested-by: Alexandre Courbot <gnurou@gmail.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit d621e8bae5ac9c67de4de90c5cded12adc8ee1e1
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:13 2014 +0100

 gpio/gpiolib-of: Create of_mm_gpiochip_remove

 Create counterpart of of_mm_gpiochip_add(). This way the modules that
 can be removable do not duplicate the cleanup code.

 Suggested-by: Alexandre Courbot <gnurou@gmail.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit c458e45045da96b4d3506ba2acab02af8c98c8c2
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Dec 17 16:51:14 2014 +0100
```

```
 gpio/xilinx: Use of_mm_gpiochip_remove

 Use the newly created of_mm_gpiochip_remove function for cleaning up
 of_mm_gpiochip_add

 Suggested-by: Alexandre Courbot <gnurou@gmail.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit f28f8eff9148eb2a3a568cc378167ba28dd2f225
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jan 13 10:41:57 2015 +0100

 gpio/Kconfig: Fix X86 arch name

 X86 Kconfig symbol is X86, not ARCH_X86.

 Fixes: c586b3075d5b47d8 (gpio/xilinx: Add support for X86 Arch)
 Reported-by: Paul Bolle <pebolle@tiscali.nl>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit f91b2dbba5fdbe2d211aa8a4b353a8d3b5126f45
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:26 2015 +0100

 gpio: mpc5200: Use of_mm_gpiochip_remove

 Since d621e8bae5ac9c67 (Create of_mm_gpiochip_remove), there is a
 counterpart for of_mm_gpiochip_add.

 This patch implements the remove function of the driver making use of
 it.

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: Sascha Hauer <s.hauer@pengutronix.de>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
```

```
commit 080440a27403c906e8ed67660344fca4b757ed8d
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:27 2015 +0100

 gpio :gpio-mm-lantiq: Use devm_kzalloc

 Replace kzalloc with the device managed devm_kzalloc

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: John Crispin <blogic@openwrt.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit f3f26c2f4b15b5b56ed87610403e4a98a13d6369
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:28 2015 +0100

 gpio: gpio-mm-lantiq: Do not replicate code

 Do not replicate code from of_mm_gpiochip_add.

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: John Crispin <blogic@openwrt.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit 68a99b187df3f4675f22a4a3f54c9d7645ab6409
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Jan 19 12:27:24 2015 +0100

 gpio: gpio-mm-lantiq: Use of_property_read_u32

 Instead of parsing manually the shadow content, use the much simpler
 helper of_property_read_u32.

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: Grant Likely <grant.likely@linaro.org>
 Cc: Rob Herring <robh+dt@kernel.org>
 Cc: John Crispin <blogic@openwrt.org>
```

```
 Cc: devicetree@vger.kernel.org
 Cc: Mark Rutland <mark.rutland@arm.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit da2382218c37fe7f8186c20b6b18aba477ec9b19
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:30 2015 +0100

 gpio: gpio-mm-lantiq: Use of_mm_gpiochip_remove

 Since d621e8bae5ac9c67 (Create of_mm_gpiochip_remove), there is a
 counterpart for of_mm_gpiochip_add.

 This patch implements the remove function of the driver making use of
 it.

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: John Crispin <blogic@openwrt.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit ff00be69fd95ea02b1274ea2ea1474727adeffd5
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:31 2015 +0100

 gpio: zevio: Use of_mm_gpiochip_remove

 Since d621e8bae5ac9c67 (Create of_mm_gpiochip_remove), there is a
 counterpart for of_mm_gpiochip_add.

 This patch implements the remove function of the driver making use of
 it.

 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: Fabian Vogt <fabian@ritter-vogt.de>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>
```

```
commit 98686d9a52eeeab83a33fca5c19448954d109458
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:32 2015 +0100

 gpio: mpc8xxx: Convert to platform device interface.

 This way we do not need to transverse the device tree manually.

 Cc: Grant Likely <grant.likely@linaro.org>
 Cc: Rob Herring <robh+dt@kernel.org>
 Cc: devicetree@vger.kernel.org
 Acked-by: Peter Korsgaard <peter@korsgaard.com>
 Acked-by: Alexandre Courbot <acourbot@nvidia.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit 257e10752c13f2698d53e5df1744f4d7e41fdfa7
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sun Jan 18 12:39:33 2015 +0100

 gpio: mpc8xxx: Use of_mm_gpiochip_remove

 Since d621e8bae5ac9c67 (Create of_mm_gpiochip_remove), there is a
 counterpart for of_mm_gpiochip_add.

 This patch implements the remove function of the driver making use of
 it.

 Cc: Linus Walleij <linus.walleij@linaro.org>
 Cc: Alexandre Courbot <gnurou@gmail.com>
 Cc: Peter Korsgaard <jacmet@sunsite.dk>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Linus Walleij <linus.walleij@linaro.org>

commit bca690db90b832a58756b30d5ff41e65881216f9
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Jan 23 17:08:33 2015 +0100

 spi/xilinx: Support for spi mode LSB_FIRST
```

```
 Hardware supports LSB_FIRST mode. Support it also in the driver.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 0240f94516964db00d0fc1d1e676f3c273710e17
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Jan 23 17:08:34 2015 +0100

 spi/xilinx: Support for spi mode LOOP

 Hardware supports LOOP mode. Support it also in the driver.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit c5d348dffa7ca1f072ff5526171a2b88bbffc24b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Jan 23 17:08:35 2015 +0100

 spi/xilinx: Simplify data read from the Rx FIFO

 The number of words in the read buffer will be exactly the same as the
 number of words written on write buffer, once the transaction has
 finished.

 Instead of cheking the rx_empty flags for every word simply save the
 number of words written by fill_tx_fifo.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 4c9a761402d780b86b9b068aba4ef8e29ed15e99
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:40 2015 +0100

 spi/xilinx: Simplify spi_fill_tx_fifo
```

```
 Instead of checking the TX_FULL flag for every transaction, find out the
 size of the buffer at probe time and use it.

 To avoid situations where the core had some data on the buffer before
 initialization, the core is reseted before the buffer size is detected

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 899929babac9d85ddd3f86b813a8b8654ab93523
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:41 2015 +0100

 spi/xilinx: Leave the IRQ always enabled.

 Instead of enabling the IRQ and disabling it for every transaction.

 Specially the small transactions (1,2 words) benefit from removing 3 bus
 accesses.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit a87cbca0acef0b1f57e4b216f1965468ee521cd3
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:42 2015 +0100

 spi/xilinx: Code cleanup

 On the transmission loop, check for remaining bytes at the loop
 condition.

 This way we can handle transmissions of 0 bytes and clean the code.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 5b74d7a3b888fd3db6dce77eb11b3d55b64f6833
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
Date:   Wed Jan 28 13:23:43 2015 +0100

 spi/xilinx: Use cached value of register

 The control register has not changed since the previous access.
 Therefore we can use the cached value and safe one bus access.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 5fe11cc09ce81b000b1deadcdec3813fcb423c8c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:44 2015 +0100

 spi/xilinx: Support cores with no interrupt

 The core can run in polling mode. In fact, the performance of the core
 is similar (or even better), due to the fact most of the spi
 transactions are just a couple of bytes and there is one irq per
 transactions.

 When an mtd device is connected via spi, reading 8MB of data produces
 more than 80K interrupts (with irq disabling, context swith....)

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit d9f5881242db8f9dc8a262c4bc613ba6fcb66bfc
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:45 2015 +0100

 spi/xilinx: Do not inhibit transmission in polling mode

 When no irq is used, there is no need to inhibit the transmission for
 every transaction. This inhibition was implemented to avoid a race
 condition with the irq handler.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>
```

```
commit f9c6ef6cfe9c16b6681607afd7e4f8379e615c4f
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:46 2015 +0100

 spi/xilinx: Support for spi mode CS_HIGH

 The core controls the chip select lines individually.

 By default, all the lines are consider active_low. After
 spi_setup_transfer, it has its real value.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 24ba5e593f391507c614f5b62065194e6593a658
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:47 2015 +0100

 spi/xilinx: Remove rx_fn and tx_fn pointer

 Simplify the code by removing the tx and and rx function pointers and
 substitute them by a single function.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit c30929415a3d7f186da888d322f93150af308287
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:48 2015 +0100

 spi/xilinx: Make spi_tx and spi_rx simmetric

 spi_rx handles the case where the buffer is null. Nevertheless spi_tx
 did not handle it, and was handled by the caller function.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>
```

```
commit d79b2d073ae4eeb79ce1aa27c96fe9e3f7582e97
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:49 2015 +0100

 spi/xilinx: Convert remainding_bytes in remaining words

 Simplify the code by using the unit used on most of the code logic.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 17aaaa80327107843f8ea6d2629ab5f733fc01aa
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:50 2015 +0100

 spi/xilinx: Convert bits_per_word in bytes_per_word

 Simplify the code by using the unit used on most of the code logic.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 99082eab63449f9dfa83d5157fa6d78bfc1b04d7
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:51 2015 +0100

 spi/xilinx: Remove iowrite/ioread wrappers

 Save a stack level and cleanup code.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit b563bfb8d76762fe5a4db1e4f983f3647d60e456
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:52 2015 +0100

 spi/xilinx: Remove remaining_words driver data variable
```

```
 The variable never leaves the scope of txrx_bufs.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 22417352f6b7f623495cc426680de75d725197df
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 13:23:54 2015 +0100

 spi/xilinx: Use polling mode on small transfers

 Small transfers generally can be accomplished faster in polling mode.
 This patch select the transfer which size is bellow the buffer size to
 be done on polling mode

 Suggested-by: Mark Brown <broonie@kernel.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit eb25f16c6f446936932f41f3ff811fbdc285bbc4
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Jan 28 20:53:39 2015 +0100

 spi/xilinx: Check number of slaves range

 The core only supports up to 32 slaves, and the chipselect function
 expects the same.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 34093cb97abe9298a19a96b1b2f1714e28f8e67f
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Feb 2 11:06:56 2015 +0100

 spi/xilinx: Fix access invalid memory on xilinx_spi_tx

 On 1 and 2 bytes per word, the transfer of the 3 last bytes will access
 memory outside tx_ptr.
```

```
 Although this has not trigger any error on real hardware, we should
 better fix this.

 Fixes: 24ba5e593f391507 (Remove rx_fn and tx_fn pointer)
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>

commit 048035bf8c7b752e9fe608ff3069e41c460efda5
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Feb 2 19:26:32 2015 +0100

 staging/unisys/visorutil/procobjecttree: Code Style

 Lines should not be over 80 characters

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4abfc1f5fb846bbc8b151aa35eae2c2fea686e89
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Feb 2 19:55:46 2015 +0100

 staging/unisys/visorutil/procobjecttree: Replace typedef

 Instead of declaring a new type, define a new struct.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b273c2c2f2d2d13dc0bfa8923d52fbaf8fa56ae8
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Feb 2 20:27:11 2015 +0100

 x86/apic: Fix the devicetree build in certain configs

 Without this patch:

   LD      init/built-in.o
```

```
   arch/x86/built-in.o: In function ‘dtb_lapic_setup’:
   undefined reference to ‘apic_force_enable’
   Makefile:923: recipe for target ’vmlinux’ failed
   make: *** [vmlinux] Error 1

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reviewed-by: Maciej W. Rozycki <macro@linux-mips.org>
 Cc: David Rientjes <rientjes@google.com>
 Cc: Jan Beulich <JBeulich@suse.com>
 Signed-off-by: Ingo Molnar <mingo@kernel.org>

commit 9555b47fab149ee23bddc842c264dd6f3b51f52d
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Thu Jan 29 15:52:02 2015 +0100

 sparc: io_64.h: Replace io function-link macros

 Function like macros cannot be assigned to function pointers. This patch
 convert the function-like macros into object-macros, that the
 precompiler will replace with the name of the final function.

 With this patch this kind of code will work:

 if (priv->mode_big_endian)
         priv.read = ioread32be;
 else
         priv.read = ioread32;

 Same approach has been taken on asm-generic/io.h

 Reported-by: kbuild test robot <fengguang.wu@intel.com>
 Fixes: 99082eab63449f9d spi/xilinx: Remove iowrite/ioread wrappers
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: David S. Miller <davem@davemloft.net>
 Signed-off-by: David S. Miller <davem@davemloft.net>

commit 8582e267e9a29ccfd2151c5d74bc2b1440dfb827
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Feb 11 13:53:14 2015 +0100
```

```
 asm/dma-mapping-common: Clarify output of dma_map_sg_attrs

 Although dma_map_sg_attrs returns 0 on error and it cannot return a
 value < 0, the function returns a signed integer.

 Most of the time, this function is used with a scatterlist structure.
 This structure uses an unsigned integer for the number of memory.

 A dma developer that has not read in detail DMA-API.txt, can wrongly
 return a value < 0 on error.

 The comment will help the driver developer, and the WARN_ON the dma
 developer.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Marek Szyprowski <m.szyprowski@samsung.com>

commit 04abab698285297115e5096b3100df1064045529
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Feb 11 13:53:15 2015 +0100

 include/dma-mapping: Clarify output of dma_map_sg

 Although dma_map_sg returns 0 on error and it cannot return a
 value < 0, the function returns a signed integer.

 Most of the time, this function is used with a scatterlist structure.
 This structure uses an unsigned integer for the number of memory.

 A dma developer that has not read in detail DMA-API.txt, can wrongly
 return a value < 0 on error.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Marek Szyprowski <m.szyprowski@samsung.com>

commit 6a9df4303b2325a7ed3c44fbf4f8878798fb2c94
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Mar 16 11:51:09 2015 +0100
```

```
 staging/sm75fb: Declare static functions as such

 This patch fixes this sparse warning

 CHECK   drivers/staging/sm750fb/ddk750_swi2c.c
 drivers/staging/sm750fb/ddk750_swi2c.c:223:6: warning: symbol
   'swI2CStart' was not declared. Should it be static?
 drivers/staging/sm750fb/ddk750_swi2c.c:234:6: warning: symbol
   'swI2CStop' was not declared. Should it be static?
 drivers/staging/sm750fb/ddk750_swi2c.c:252:6: warning: symbol
   'swI2CWriteByte' was not declared. Should it be static?
 drivers/staging/sm750fb/ddk750_swi2c.c:320:15: warning: symbol
   'swI2CReadByte' was not declared. Should it be static?
 drivers/staging/sm750fb/ddk750_swi2c.c:361:6: warning: symbol
   'swI2CInit_SM750LE' was not declared. Should it be static?

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 82fdb8dd11851b4c12da78c4626740c83bc3fc2d
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Mar 16 23:01:44 2015 +0100

 staging/goldfish/goldfish_audio: Fix annotations

 dmam_alloc_coherent does not return a __iomem pointer.
 here is its prototype:

 void * dmam_alloc_coherent(struct device *dev, size_t size,
                            dma_addr_t *dma_handle, gfp_t gfp)

 This fixes these sparse warnings:

 drivers/staging/goldfish/goldfish_audio.c:134:43: warning: incorrect
 type in argument 2 (different address spaces)
 drivers/staging/goldfish/goldfish_audio.c:134:43:    expected void const
 *from
 drivers/staging/goldfish/goldfish_audio.c:134:43:    got char [noderef]
```

```
 <asn:2>*read_buffer
 drivers/staging/goldfish/goldfish_audio.c:167:36: warning: incorrect
 type in argument 1 (different address spaces)
 drivers/staging/goldfish/goldfish_audio.c:167:36:    expected void *to
 drivers/staging/goldfish/goldfish_audio.c:167:36:    got char [noderef]
 <asn:2>*[assigned] kbuf
 drivers/staging/goldfish/goldfish_audio.c:296:27: warning: incorrect
 type in assignment (different address spaces)
 drivers/staging/goldfish/goldfish_audio.c:296:27:    expected char
 [noderef] <asn:2>*buffer_virt
 drivers/staging/goldfish/goldfish_audio.c:296:27:    got void *

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit a96aa64cb5723d941de879a9cd1fea025d6acb1b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Mar 30 10:45:59 2015 -0700

 leds/led-class: Handle LEDs with the same name

 The current code expected that every LED had an unique name. This is a
 legit expectation when the device tree can no be modified or extended.
 But with device tree overlays this requirement can be easily broken.

 This patch finds out if the name is already in use and adds the suffix
 _1, _2... if not.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reported-by: Geert Uytterhoeven <geert@linux-m68k.org>
 Signed-off-by: Bryan Wu <cooloney@gmail.com>

commit b08d8d26f4520187e09191f0c45374383b5a74eb
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Mar 20 10:30:46 2015 -0300

 [media] media/v4l2-ctrls: volatiles should not generate CH_VALUE
```

```
 Volatile controls should not generate CH_VALUE events.

 Set has_changed to false to prevent this happening.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit b6e5b8f1a90230f922de8af59cdaf1ad83e1ac1e
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Mar 20 10:45:43 2015 -0300

 [media] media: New flag V4L2_CTRL_FLAG_EXECUTE_ON_WRITE

 Create a new flag that represent controls which its value needs to be
 passed to the driver even if it has not changed.

 They typically represent actions, like triggering a flash or clearing an
 error flag. So writing to such a control means some action is executed.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit ef66c0ca09595d72f058b11bd7b542b0d95a1350
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Mar 20 11:21:28 2015 -0300

 [media] media/v4l2-ctrls: Add execute flags to write_only controls

 Any control that sets FLAG_WRITE_ONLY should OR it with
 FLAG_EXECUTE_ON_WRITE.

 So we can keep the current meaning of WRITE_ONLY.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>
```

```
commit 45f014c5264f5e68ef0e51b36f4ef5ede3d18397
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Mar 20 10:55:37 2015 -0300

 [media] media/v4l2-ctrls: Always execute EXECUTE_ON_WRITE ctrls

 Any control with V4L2_CTRL_FLAG_EXECUTE_ON_WRITE set should return
 changed == true in cluster_changed.

 This forces the value to be passed to the driver even if it has not
 changed.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit acd2b672b45e557829c1f080e46457586aab9d74
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Mar 20 11:13:14 2015 -0300

 [media] media/Documentation: New flag EXECUTE_ON_WRITE

 Document new flag V4L2_CTRL_FLAG_EXECUTE_ON_WRITE, and the new behavior
 of CH_VALUE event on VOLATILE controls.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 3ca7bf8756a0426e642446ae35df31a29a1b1108
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Apr 15 16:17:34 2015 -0700

 CREDITS: add Ricardo Ribalda Delgado

 Add personal details to CREDITS file.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
```

```
 Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

commit 01c7e5a50cc16174b7ccfdbcf32e9bb136f277ba
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Apr 15 16:17:37 2015 -0700

 .mailmap: add Ricardo Ribalda

 Work and Home computer had different settings in the mail client.  Some
 contributions appear as Ricardo Ribalda, others as
 Ricardo Ribalda Delgado (and one as just Ricardo).

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
 Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

commit 18b3b3b8edb136d90e5d16d986ce9ae7eaeb0776
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon May 4 05:07:29 2015 -0300

 [media] media/vivid: Add support for Y16 format

 Support for V4L2_PIX_FMT_Y16, a 16 bit greyscale format.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit ed1bc664daf58af7da47581fe43d42b95a1de65a
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon May 4 05:07:32 2015 -0300

 [media] media/vivid: Code cleanout

 Remove code duplication by merging two cases in a switch.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>
```

```
commit 6a5d77cbf26078e9ee8f88c7d77fe0a7fa4a0364
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Apr 29 09:00:45 2015 -0300

 [media] media/videobuf2-dma-sg: Fix handling of sg_table structure

 When sg_alloc_table_from_pages() does not fail it returns a sg_table
 structure with nents and nents_orig initialized to the same value.

 dma_map_sg returns the number of areas mapped by the hardware,
 which could be different than the areas given as an input.
 The output must be saved to nent.

 The output of dma_map, should be used to transverse the scatter list.

 dma_unmap_sg needs the value passed to dma_map_sg (nents_orig).

 sg_free_tables uses also orig_nent.

 This patch fix the file to follow this paradigm.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reviewed-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 60a471923869394a47c10262d080286eb01451b5
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Apr 29 09:00:46 2015 -0300

 [media] media/videobuf2-dma-contig: Save output from dma_map_sg

 dma_map_sg returns the number of areas mapped by the hardware,
 which could be different than the areas given as an input.
 The output must be saved to nent.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reviewed-by: Marek Szyprowski <m.szyprowski@samsung.com>
```

```
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit ba81c6ed3dd4717439940cd5f60b152ea924093d
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Apr 29 09:00:47 2015 -0300

 [media] media/videobuf2-dma-vmalloc: Save output from dma_map_sg

 dma_map_sg returns the number of areas mapped by the hardware,
 which could be different than the areas given as an input.
 The output must be saved to nent.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Reviewed-by: Marek Szyprowski <m.szyprowski@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 2e5e435fb4fdcc64db49e903baddb1ea8827385e
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon May 4 05:07:30 2015 -0300

 [media] media/v4l2-core: Add support for V4L2_PIX_FMT_Y16_BE

 16 bit greyscale format, structured in Big Endian. Such a format can be
 converted into a PMN image just by adding a header.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit b0ce23f065744ba136e6b059369f51e5ae1fb769
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Mon May 4 05:07:31 2015 -0300

 [media] media/vivid: Add support for Y16_BE format

 Support for V4L2_PIX_FMT_Y16_BE, a 16 bit big endian greyscale format.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 36d4b29260753ad78b1ce4363145332c02519adc
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 26 09:31:23 2015 +0200

 base/platform: Only insert MEM and IO resources

 platform_device_del only checks the type of the resource in order to
 call release_resource.

 On the other hand, platform_device_add calls insert_resource for any
 resource that has a parent.

 Make both code branches balanced.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit e50e69d1ac4232af0b6890f16929bf5ceee81538
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 26 09:31:24 2015 +0200

 base/platform: Continue on insert_resource() error

 insert_resource() can fail when the resource added  overlaps
 (partially or fully) with another.

 Device tree and AMBA devices may contain resources that overlap, so they
 could not call platform_device_add (see 02bbde7849e6 ('Revert "of:
 use platform_device_add"'))"

 On the other hand, device trees are released using
 platform_device_unregister(). This function calls platform_device_del(),
 which calls release_resource(), that crashes when the resource has not
 been added with with insert_resource. This was not an issue when the
 device tree could not be modified online, but this is not the case
```

```
 anymore.

 This patch let the flow continue when there is an insert error, after
 notifying the user with a dev_err(). r->parent is set to NULL, so
 platform_device_del() knows that the resource was not added, and
 therefore it should not be released.

 Acked-by: Rob Herring <robh@kernel.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit b6d2233f2916fa9338786aeab2e936c5a07e4d0c
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 26 09:31:25 2015 +0200

 of/platform: Use platform_device interface

 of_platform_device_create_pdata() was using of_device_add() to create
 the devices, but of_platform_device_destroy was using
 platform_device_unregister() to free them.

 of_device_add(), do not call insert_resource(), which initializes the
 parent field of the resource structure, needed by release_resource(),
 called by of_platform_device_destroy(). This leads to a NULL pointer
 deference.

 Instead of fixing the NULL pointer deference, what could hide other bugs,
 this patch, replaces of_device_add() with platform_device_data().

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Rob Herring <robh@kernel.org>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 6d9d4b1469b0d9748145e168fc9ec585e1f3f4b0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue May 26 09:31:26 2015 +0200

 base/platform: Remove code duplication
```

```
 Failure path of platform_device_add was almost the same as
 platform_device_del. Refactor same code in a function.

 Acked-by: Rob Herring <robh@kernel.org>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 4366dfef377034e97d5b35677b7a1ebb1f1ce6dc
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Wed Jun 10 10:38:29 2015 -0300

 [media] media/v4l2-ctrls: Code cleanout validate_new()

 We can simplify the code removing the if().

 v4l2_ctr_new sets ctrls->elems to 1 when !ctrl->is_ptr.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 747d481df3b9b6ef46ccb233d74c8d93ec4819e6
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:07 2015 -0300

 [media] media/i2c/adv7343: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 260faaa490d29cd2ad3cad54fbda0ede406c818e
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:08 2015 -0300
```

```
 [media] media/i2c/adv7393: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 6ceea1f7d021492ed13609fa1cf401484dae0e0d
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:09 2015 -0300

 [media] media/i2c/cs5345: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 3cfa008e3cafa90684f84abbfb995cec437a61ea
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:10 2015 -0300

 [media] media/i2c/saa717x: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit fc1a33fed0274efefbfc5f463a69efbaee4d989f
```

```
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:12 2015 -0300

 [media] media/i2c/tda7432: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 8b198c6b8b65ef39c3652972c36a43e04b4482dd
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:13 2015 -0300

 [media] media/i2c/tlv320aic23: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit e5b40d2e11c3e2ed60ecb70313765d48c7786448
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:14 2015 -0300

 [media] media/i2c/tvp514x: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
```

```
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit f5f24bc3267b3651e8bb026a5669dfe6855ffe63
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:15 2015 -0300

 [media] media/i2c/tvp7002: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit c2527967c5795815d3422f147af767293b3806eb
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:16 2015 -0300

 [media] i2c/wm8739: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 59617e74e20dac9523e34f5461162b19347dacf8
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:17 2015 -0300

 [media] pci/ivtv/ivtv-gpio: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.
```

```
 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 387a69243890a51be023aef12d0fdcae343f43f2
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:18 2015 -0300

 [media] media/radio/saa7706h: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 4c34cc5e0f99ced4c7a11d49007bf7a90e2fae7a
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Fri Jun 12 13:31:11 2015 -0300

 [media] media/i2c/sr030pc30: Remove compat control ops

 They are no longer used in old non-control-framework
 bridge drivers.

 Reported-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Sylwester Nawrocki <s.nawrocki@samsung.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>
 Signed-off-by: Mauro Carvalho Chehab <mchehab@osg.samsung.com>

commit 92a5f11a1a3357964c30c34d55dc55152315b81b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jun 16 10:59:36 2015 +0200

 serial/8250_fintek: Use private data structure
```

```
 Save the port index and the line id in a private structure.

 Reported-by: Peter Hong <peter_hong@fintek.com.tw>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 017bec38c08fc954fb788fe69cf9e55f0d4910e7
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jun 16 10:59:37 2015 +0200

 serial/8250_fintek: Support for multiple base_ports

 Fintek chip can be connected at address 0x4e and also 0x2e.
 Add some logic to find out the address of the chip.

 Reported-by: Peter Hong <peter_hong@fintek.com.tw>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit dae77f757950c273500cc72262a094f290578bef
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jun 16 10:59:38 2015 +0200

 serial/8250_fintek: Support for chip_ip 0x0501

 There are some chips with the same interface but different chip ip.

 Reported-by: Peter Hong <peter_hong@fintek.com.tw>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit ce8c267e2ef9180d21f6915f21874898c2ac0ffb
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jun 16 10:59:39 2015 +0200
```

```
 serial/8250_fintek: Support keys different than default

 Chip can be configured to use entry key different than 0x77. Try all the
 valid keys until one gives out the right chip id.

 Reported-by: Peter Hong <peter_hong@fintek.com.tw>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 29d58642f1a090ad574d14872b610b365b2d023b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jun 16 10:59:40 2015 +0200

 serial/8250_fintek: Support for any io address.

 Fintek chip can be configured for io addresses different than the standard.

 Query the chip for the configured addresses and try to match it with the
 pnp address.

 Reported-by: Peter Hong <peter_hong@fintek.com.tw>
 Reviewed-by: Alan Cox <alan@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 1eacbfb06df9d37b743d0eedbeb9fd4f71190c98
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Jun 15 16:36:07 2015 +0200

 serial/uartlite: Let it build on any arch with IOMEM

 Being a soft core, it can be located not only on PPC or Microblaze
 platforms.

 Since the driver already does endianness detection we only need to
 change the Kconfig to use it in other arches.

 This is also done in other softcores as xilinx-spi.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

commit 16ea9b8ac45bf11d48af6013283e141e8ed86348
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 12 18:04:04 2015 +0200

 spi/spi-xilinx: Fix mixed poll/irq mode

 Once the module process a transfer in irq mode, the next poll transfer
 will not work because the transmitter is left in inhibited state.

 Fixes: 22417352f6b7f623 (Use polling mode on small transfers)
 Reported-by: Edward Kigwana <ekigwana@scires.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>
 Cc: stable@vger.kernel.org

commit 74346841e6f5df5f7b83d5904435d273c507dba6
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Aug 13 16:09:28 2015 +0200

 spi/spi-xilinx: Fix spurious IRQ ACK on irq mode

 The ACK of an inexistent IRQ can trigger an spurious IRQ that breaks the
 txrx logic. This has been observed on axi_quad_spi:3.2 core.

 This patch only ACKs IRQs that have not been Acknowledge jet.

 Reported-by: Edward Kigwana <ekigwana@scires.com>
 Tested-by: Edward Kigwana <ekigwana@scires.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Mark Brown <broonie@kernel.org>
 Cc: stable@vger.kernel.org

commit e5b5a61fcb3743f1dacf9e20d28f48423cecf0c1
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Jul 31 13:36:21 2015 +0200
```

```
leds/led-class: Add missing put_device()

Devices found by class_find_device must be freed with put_device().
Otherwise the reference count will not work properly.

Fixes: a96aa64cb572 ("leds/led-class: Handle LEDs with the same name")
Reported-by: Alan Tull <delicious.quinoa@gmail.com>
Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Signed-off-by: Jacek Anaszewski <j.anaszewski@samsung.com>
```

B

CONTRIBUTIONS TO U-BOOT

The FPGA system, described in chapter 4, required a bootloader capable of reading the kernel from a flash memory or a network connection, as well as serve as a tool for the board bring-up.

Das U-Boot, or just U-Boot, is an Open Source bootloader widely adopted by the industry. It has support for many architectures and boards and has a well tested network stack.

When the system was developed, there was no upstream support for the Xilinx Boards, nor its embedded PowerPC CPUs (ppc440 and ppc405). As a result of this Thesis, U-boot was extended to support Xilinx designs based on PowerPC embedded cores.

Due to this contributions, the author of this Thesis is the maintainer for the following boards: Avnet V5FX30TEVAL, Xilinx ML507, Xilinx PPC405-GENERIC and Xilinx PPC440-GENERIC.

The current list of contributions can be found on the Official U-Boot Repository.

```
commit d865fd09809a3a18669f35f970781820af40e4de
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Thu Jul 17 11:44:12 2008 +0200

 ppc4xx: CPU PPC440x5 on Virtex5 FX

 -This patchs gives support for the embbedded ppc440
  on the Virtex5 FPGAs
 -interrupts.c divided in uic.c and interrupts.c
```

```
 -xilinx_irq.c for xilinx interrupt controller
 -Include modifications propossed by  Stefan Roese

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Acked-by: Stefan Roese <sr@denx.de>

commit 086511fc96a8a9bb56e5e19a3d84c40f4dba80cc
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Thu Jul 17 12:47:09 2008 +0200

 ppc4xx: ML507 Board Support

 The Xilinx ML507 Board is a Virtex 5 prototyping board that includes,
         among others:
         -Virtex 5 FX FPGA (With a ppc440x5 in it)
         -256MB of SDRAM2
         -32MB of Flash
         -I2C Eeprom
         -System ACE chip
         -Serial ATA connectors
         -RS232 Level Conversors
         -Ethernet Transceiver

 This patch gives support to a standard design produced by EDK for this
 board: ppc440, uartlite, xilinx_int and flash

 - Includes Changes propossed by Stefan Roese and Michal Simek

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Acked-by: Stefan Roese <sr@denx.de>

commit f13f64cf42d5abec3e0f920233f6a7a61e7ae494
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Wed Jul 16 16:22:32 2008 +0200

 serial_xuartlite.c: fix compiler warnings

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Acked-by: Grant Likely <grant.likely@secretlab.ca>
```

```
commit 01a004313c5ec2d128b611df4c208b1b0d3c3fb4
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Jul 21 20:30:07 2008 +0200

 ppc4xx: ML507: U-Boot in flash and System ACE

 This patch allows booting from FLASH the ML507 board by Xilinx.
 Previously, U-Boot needed to be loaded from JTAG or a Sytem ACE CF

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit a8a16af4d59d14cc1c1187c10aaad80d6b8394b5
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Tue Jul 29 17:16:10 2008 +0200

 ppc4xx: ML507: Use of get_ram_size in board ml507

 - Change suggested by WD

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 9246f5ecfd353ae297a02ffd5328402acf16c9dd
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Wed Jul 30 12:39:28 2008 +0200

 ppc4xx: ML507: Environment in flash and MTD Support

 - Relocate the location of U-Boot in the flash
 - Save the environment in one sector of the flash memory
 - MTD Support

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit d0039d4ed275e6ca09fb417895024ad02be118c4
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
```

```
Date:   Wed Jul 23 19:10:14 2008 +0200

 Add support for ADT7460 I2C monitor chip

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>

commit e07f4a8033b6270b8103049adb6456f660ff4a89
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Sep 1 13:09:39 2008 -0400

 ppc44x: Unification of virtex5 pp440 boards

 This patch provides an unificated way of handling xilinx v5
 ppc440 boards.

 It unificates 3 different things:

 1) Source code
 A new board called ppc440-generic has been created. This board
 includes a generic tlb initialization (Maps the whole memory into
 virtual) and defines board_pre_init, checkboard, initdram and
 get_sys_info weakly, so, they can be replaced by specific functions.

 If a new board needs to redefine any of the previous functions
 (specific initialization) it can create a new directory with the
 specific initializations needed. (see the example ml507 board).

 2) Configuration file
 Common configurations are located under configs/xilinx-ppc440.h,
 this header file interpretes the xparameters file generated by
 EDK and configurates u-boot in correspondence. Example: if there
 is a Temac, allows CMD_CONFIG_NET
 Specific configuration are located under specific configuration
 file. (see the example ml507 board)

 3) Makefile
 Some work has been done in order to not duplicate work in the
 Main Makefile. Please see the attached code.
```

```
 In order to support new boards they can be implemented in the
 next way:

 a) Simple Generic Board  (90% of the time)
 Using EDK generates a new xparameters.h file, replace
 ppc440-generic/xparameters.h  and run make
 xilinx-ppc440-generic_config && make

 b) Simple Boards with special u-boot parameters (9 % of the time)
 Create a new file under configs for it (use ml507.h as example)
 and change your paramaters. Create a new Makefile paragraph and
 compile

 c) Complex boards (1% of the time)
 Create a new folder for the board, like the ml507

 Finally, it adds support for the Avnet FX30T Evaluation board,
 following the new generic structure:

 Cheap board by Avnet for evaluating the Virtex5 FX technology.

 This patch adds support for:
  - UartLite
  - 16MB Flash
  - 64MB RAM

 Prior using U-boot in this board, read carefully the ERRATA by
 Avnet to solve some memory initialization issues.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 2b22d608f370565c87f55928b524207031419c11
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Wed Jul 30 12:39:29 2008 +0200

 loads: allow negative offsets

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
```

```
commit 0817d688f307ee2c0598e79175c94a40ce90337b
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Sun Sep 7 17:10:27 2008 -0400

 Remove gap fill in srec object v2

 SREC files do not need gap fill: The load address is specified
 in the file. On the other hand, it can't be avoided in a .bin
 object. It has no information about memory location.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>

commit 880f6a5d7596f42db5ff8803b797b78ec5b146e0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Tue Sep 9 10:00:33 2008 -0400

 ppc4xx: ppc440-generic-ALL: Fix out of tree build v2

 This patch solves the problems compiling ml507, v5fx30teval and
 ppc440-generic out of tree.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>

commit 4bed9deebbd7ee6f0ba746b44d47a922156f7404
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Wed Sep 10 17:44:30 2008 -0400

 ppc4xx: Fix in-tree build for ppc440-generic boards

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 2bec498ed1164a58cd8437b561bdc4551d69f9bf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Thu Sep 11 19:41:25 2008 -0400

 ppc4xx: Fix compilation of v5fx30teval_flash
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 61737c59a3285f6fadf96a5836879898c04ec28d
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Thu Sep 11 19:41:26 2008 -0400

 ppc4xx: Add .gitignore file to xilinx-ppc440 boards

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 1f4d53260ec6f8f122aed75cce7c757d97a551e0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Tue Oct 21 18:29:46 2008 +0200

 ppc4xx: Generic architecture for xilinx ppc405(v3)

 As "ppc44x: Unification of virtex5 pp440 boards" did for the
 xilinx ppc440 boards, this patch presents a common architecture
 for all the xilinx ppc405 boards.

 Any custom xilinx ppc405 board can be added very easily with no
 code duplicity.

 This patch also adds a simple generic board, that can be used on
 almost  any design with xilinx ppc405 replacing the file
 ppc405-generic/xparameters.h

 This patch is prepared to work with the latest version of EDK
 (10.1)

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit cc2dc9b08cf7c09f9f237f8cb9303f11603d4fb0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Oct 27 12:35:59 2008 +0100
```

```
 ppc4xx: Merge xilinx-ppc440 and xilinx-ppc405 cfg

 Xilinx ppc440 and ppc405 have many similarities. This patch merge
 the config files of both infrastuctures

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit dbd33614404b65aa441c5620c3dbd560c4460c09
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Apr 27 09:13:31 2009 +0200

 ubifs: BUG realpath string must be ended with NULL

 If the memory used to copy the link_make is "dirty" the string
 wont be ended with NULL, throwing out multiple memory bugs.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
 Acked-by: Stefan Roese <sr@denx.de>

commit 35f6a943f7d92145d607c1d55f5c2e2eae5be630
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Apr 27 18:33:32 2009 +0200

 lib_generic: gunzip: New function zunzip

 Separate gunzip in

 gunzip: Find the end of the header and call zunzip.
 zunzip: Inflate gunzip block without header.

 UBI fs blocks can be compresed in lzo, zlib or no-compression.
 The current implementation of u-boot supported all the compressions
 but there was a bug in the implementation of the zlib blocks.

 UBIFS's Zlib blocks do not have header but they were compressed
 using gunzip, a function used to decompress gunzip files/sectors
 with a header.
```

```
 This patch adds a new function zunzip that uncompress a zlib block
 with no header.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>

commit c1a0fd5f2864e9d381f4a3dc948942cac974e89a
Author: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>
Date:   Mon Apr 27 18:33:33 2009 +0200

 ubifs: BUG: Blocks commpressed with zlib

 Blocks compressed with zlib dont have the full gzip header.

 Without this patch, block compressed with zlib cannot be readed!

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@uam.es>

commit 64b68178489b6845bcf460e9c6e618cb81740faf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Dec 2 15:02:35 2010 +0100

 ubifs.c: BUG: Error following links

 The link_name variable is declared inside the if block and it is used
 outside it through the name pointer.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit d20b9991154241466802ceb17169dc8b5f7e58df
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Dec 7 14:27:56 2010 +0100

 xilinx-ppc4xx-generic: Use common u-boot.lds

 Use common ppc4xx linker script for xilinx ppc440 and ppc405 related
 boards.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
 Signed-off-by: Stefan Roese <sr@denx.de>

commit 7e4c3a41ce2e9ace8bbca125a81f7397ba91e536
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri Dec 24 01:38:10 2010 +0100

 xilinx-ppc4xx-generic: Fix Makefile to work with MAKEALL

 config.mk only mkdirs $(obj), but we have objects shared with other
 boards located on other dirs.

 This patch mkdirs the needed dirs for the xlnx-generic boards.

 Signed-off-by: Stefan Roese <sr@denx.de>

commit d3da7225526d996493fd45ec3d259c467db9a9d0
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Jan 12 10:14:42 2011 +0100

 xilinx_ppc_boards: Change address of RESET_VECTOR

 Old address of RESET_VECTOR were overwritten by the bss sector, making
 impossible its run from xmd.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Stefan Roese <sr@denx.de>

commit a560ad7083736fa147137eda2b04fd98b159c63f
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue May 12 16:20:28 2015 +0200

 doc/README.generic-board: Trivial spell check

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>

commit d67b72d56359f2a240a2014c7b18853de3dabf37
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue May 12 16:20:29 2015 +0200
```

```
ppc: xilinx-ppc: Move to generic board support

Generic board support seems to work just fine. Tested on ml507 with
bitstream generated on the latest ISE software.

Tested-by: Georg Schardt <schardt@team-ctech.de>
Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

C

OTHER CONTRIBUTIONS TO OPEN SOURCE PROJECTS

This chapters lists the rest of accepted contributions to Open Source projects.

Contents

C.1 VIDEO4LINUX2 UTILS - LIBV4L2

Libv4l2 is a low level library, and reference application for Video4Linux2. As a result of this Thesis, the following formats were supported by the library Y16_BE, RGB32 and BGR32. All the Open Source projects that use libv4l2 to access Video4Linux2 devices can now support these formats seamlessly.

The current list of contributions can be found on the LinuxTV Git Repository.

```
commit 3ae390e54a0ba627c9e74953081560192b996df4
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Jul 30 14:59:23 2013 +0200

 v4l2_compliance: -EINVAL is expected when ret is not 0

 Otherwise the driver can never return a register

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>

commit 45ed09604578158c9c86327f678a68f854b5dba3
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sat Aug 3 00:42:51 2013 +0200

 libv4lconvert: Support for Y16 pixel format

 This patch adds support for V4L2_PIX_FMT_Y16 format.

 Acked-by: Hans de Goede <hdegoede@redhat.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Gregor Jasny <gjasny@googlemail.com>

commit 4ebca00f8ad72c5c12020963d6787b5a321a7038
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Sat Aug 3 00:42:52 2013 +0200

 libv4lconvert: Support for RGB32 and BGR32 format
```

```
 This patch adds support for V4L2_PIX_FMT_RGB32 and V4L2_PIX_FMT_BGR32
 formats.

 Acked-by: Hans de Goede <hdegoede@redhat.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Gregor Jasny <gjasny@googlemail.com>

commit cffce3dd797ff3b58e4c2ea77c2127871c82d3bf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Mar 26 17:17:34 2015 +0100

 libv4lconvert: Fix support for Y16 pixel format

 Y16 is a little-endian format. The original implementation assumed that
 it was big-endian.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans de Goede <hdegoede@redhat.com>

commit f7dc1893b19e1de233a9a2e8d7b475660b1d3487
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue May 19 11:03:04 2015 +0200

 qv4l2: gl: Add support for V4L2_PIX_FMT_Y16

 Add support for a 16 bit wide greyscale format.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>

commit 75cdf2926c52892187ab179cae40c646ceef658f
Author: Ricardo Ribalda <ricardo.ribalda@gmail.com>
Date:   Tue May 19 11:03:05 2015 +0200

 qv4l2: gl: Add support for V4L2_PIX_FMT_Y16_BE

 Add support for a 16 bit wide greyscale format in big endian.
```

```
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans Verkuil <hans.verkuil@cisco.com>

commit b11a7ac323f6c30fa9eb772965880833f367c828
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon May 18 17:07:20 2015 +0200

 libv4lconvert: Add support for V4L2_PIX_FMT_Y16_BE

 Y16_BE is the big-endian version of Y16.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Hans de Goede <hdegoede@redhat.com>
```

C.2 OPENEMBEDDED / YOCTOPROJECT

OpenEmbedded is a build framework and a meta-distributition. The contributions from this Thesis to the project improve the support for .deb packages types, fix builds of modern kernels, improve support for gstreamer1.0, add support for AMD drivers, add a recipe for OpenCV 3.0 and implement some bug fixes for OpenCV 2.

The current list of contributions can be found on the OpenEmbedded Git Repository.

```
commit a08eacc6d821d6946b23a99bca5abf785875b1cf
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Tue Mar 11 11:07:35 2014 +0100

 package_deb: Map TARGET_ARCH x86_64 to DPKG_ARCH amd64

 Without this patch packages are generated as x86_64. Which cannot be
 installed by default.


 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Saul Wold <sgw@linux.intel.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>
```

```
commit 383e6c7d5fa1f6f02b50155a77d7c82237c11ba9
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Mar 13 14:27:03 2014 +0100

 apt-native: Install apt-ftparchive

 apt-ftparchive is needed to create a Release file compatible with
 SecureApt.

 It is also a more efficient replacement of dpkg-scanpackages.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

commit c9899a7605f15f7f1ae30c4624d53c7da825b00a
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Mar 17 09:30:37 2014 +0000

 package_manager: Use apt-ftparchive for deb packages

 Use apt-ftparchive to create a Release file compatible with SecureApt.

 apt-ftparchive is also a more efficient replacement of
 dpkg-scanpackages:

 root@neopili:~/curro/qtec/qt5022/build-qt5022-cesium/build/tmp/deploy/deb/
 bobcat_64# time PSEUDO_UNLOAD=1 apt-ftparchive packages . >/tmp/kkk
 real 0m26.873s
 user 0m20.968s
 sys 0m1.212s

 root@neopili:~/curro/qtec/qt5022/build-qt5022-cesium/build/tmp/deploy/deb/
 bobcat_64# time PSEUDO_UNLOAD=1 dpkg-scanpackages . >/tmp/kkk
 dpkg-scanpackages: info: Wrote 6022 entries to output Packages file.
 real 0m59.721s
 user 0m16.668s
 sys 0m11.164s
```

```
 apt-ftparchive is not compatible with libpseudo. The calls to ftw()
 returns the path in absolute format instead of relative. This
 produces
 wrong Packages and Release files.

 ie:
 MD5Sum:
 d20227a958f6870137ce0e41b7b84307             1453
 deploy/deb/all/Release

 This is why it is called with PSEUDO_UNLOAD.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Saul Wold <sgw@linux.intel.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

commit 36cba6e00d76462e4ae314dd2af0b47472835538
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed May 7 11:20:20 2014 +0200

 package_manager: Fix Argument list too long

 Function buildhistory_list_installed_image fails with error "Argument
 list too long". This patch uses a temporal file to pass the package list
 to opkg-query-helper.py

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Saul Wold <sgw@linux.intel.com>

commit 26314886c3712f980ccc589b014a8f1802193b56
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed May 7 13:23:35 2014 +0200

 package_manager: Fix NoneType Object on do_populate_sdk

 PACKAGE_EXCLUDE can be not defined or empty, leading to a build error.
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Saul Wold <sgw@linux.intel.com>
```

```
commit de8278300eca1e29747be0a0f94787ff90926598
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Nov 3 11:13:50 2014 +0100

 debianutils: Add recipe from meta-oe

 This recipe is a running dependency of recipe apt

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Ross Burton <ross.burton@intel.com>

commit 40dd71a4e0beade84ecd686559243a10e55c3a2d
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Nov 3 11:13:51 2014 +0100

 apt: Add missing running depency debianutils

 apt-file calls run-parts with options --list and --regex:

 busybox implementation of run-parts does not support --regex. And --list
 option is not enabled on yocto busybox configuration.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Ross Burton <ross.burton@intel.com>

commit 83643dc4edb9c7656726302b92fb22d1c8652dac
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Feb 9 13:58:28 2015 +0100

 cryptodev-module: Fix build on kernel v3.19

 Kernel commit f938612dd97d481b8 removes the get_unused_fd_macro().
 This patch replaces the macro with its output.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Ross Burton <ross.burton@intel.com>

commit 81636555fa7f18407efc172c0d5b9f466b2d4014
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
Date:   Fri May 8 17:04:13 2015 +0200

 gstreamer1.0: convert GSTREAMER_1_DEBUG to PACKAGECONFIG

 It is more elegant

 Suggested-by: Ross Burton <ross.burton@intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

commit b964917c7c75c637f64389363d566f57487a14da
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri May 8 17:04:14 2015 +0200

 gstreamer1.0: Make check selectable via PACKAGECONFIG

 This way, this configuration can be easily changed via .bbappend file
 without having to re-define the whole EXTRA_OECONF.

 With --disable-check  libgstcheck is not build.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

commit 350c488883ccf5cc8ccfdd36f93052a8cd43d4b6
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Fri May 8 15:36:01 2015 +0200

 xserver-xorg: Make xinerama selectable via PACKAGECONFIG

 If xinerara can be selected via PACKAGECONFIG, xserver-xorg recipe can
 be easier modified via bbappend.

 xinerama is needed by the fglrx driver (from AMD).

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

commit 19eaf4a59f4545e049f525d0b0446a9c08d18f0f
```

```
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed May 20 13:44:18 2015 -0500

 xserver-nodm: Support reboot from inside X

 If reboot was called from inside the Xserver there could happen a race
 condition where chvt would never end, and therefore the whole system was
 stalled.

 The user could not recover the system by ssh the machine or using the
 keyboard.

 Running chvt in background fixes the issue.

 Also move sleep 1s inside stop to give time for killproc xinit for
 fix issue when chvt 1 don't work because X server is still running.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Aníbal Limón <anibal.limon@linux.intel.com>
 Signed-off-by: Richard Purdie <richard.purdie@linuxfoundation.org>

 commit e5abd3c3fb72556dd6fdce126fe7b81ed1285401
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Thu Aug 20 12:10:52 2015 +0200

 Opencv: Add OpenCV 3.0

 -Support for new PACKAGECONFIGS
 -Merge with opencv-samples

 Since it is not backward compatible with 2.x and cannot be installed in
 parallel it has a DEFAULT_PREFERENCE of -1.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Otavio Salvador <otavio@ossystems.com.br>
 Signed-off-by: Martin Jansa <Martin.Jansa@gmail.com>

commit f584699a45556531f872cc577957b71cc32765ed
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
```

```
Date:   Wed Aug 19 17:15:25 2015 +0200

 opencv_2.4: Update HOMEPAGE

 Old link is dead

 ricardo@neopili:~$ wget http://opencv.willowgarage.com/wiki/
 --2015-08-19 17:12:44--  http://opencv.willowgarage.com/wiki/
 Resolving opencv.willowgarage.com (opencv.willowgarage.com)...
 70.35.54.199
 Connecting to opencv.willowgarage.com
 (opencv.willowgarage.com)|70.35.54.199|:80... failed: Connection timed
 out.
 Retrying.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Martin Jansa <Martin.Jansa@gmail.com>

commit 2a3c0b188d8cebbd59db905916d23b8d98b2ec43
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 19 17:05:45 2015 +0200

 opencv_2.4: Remove unused INSANE_SKIP line

 Credits-to: Paul Eggleton <paul.eggleton@linux.intel.com>
 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Martin Jansa <Martin.Jansa@gmail.com>

commit 223e3d1d687178820a841582e9f7ba1716d80009
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Aug 19 17:05:44 2015 +0200

 opencv_2.4: Update LICENSE

 LICENSE file at the root of the opencv repo says:

 License Agreement
 For Open Source Computer Vision Library
 (3-clause BSD License)
```

```
Credits-to: Paul Eggleton <paul.eggleton@linux.intel.com>
Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Signed-off-by: Martin Jansa <Martin.Jansa@gmail.com>
```

C.3 CLPEAK

Clpeak is a benchmark tool for OpenCL. The contributions to the project allow the cross compilation of the application and accelerates the application by replacing logarithmic float operations with binary logic.

The pull request conversation can be read at GitHub, and the list of contributions at the Clpeak Repository.

```
commit c0cd61fd5446d20b5f4412910d7b891c46a3c04f
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Nov 3 16:34:36 2014 +0100

 CMakeLists: Remove -march=native options in case of cross compilation

 This option can break cross-compilations of the program

 In my test yocto CFLAGS -march and -mtune were ignored due to
 the -march=native.

 root@qt5022:~# clpeak
 Platform: AMD Accelerated Parallel Processing
   Device: Loveland
     Driver version  : 1598.5 (Linux x64)
     Compute units   : 2
     Clock frequency : 507 MHz
 Illegal instruction

 [  822.855887] traps: clpeak[1311] trap invalid opcode ip:405045
         sp:7fffd2429260 error:0 in clpeak[400000+20000]
```

```
 1  0x000000000040cbba in clPeak::runGlobalBandwidthTest
 2  0x0000000000406992 in clPeak::runAll() ()
 3  0x0000000000404bfe in main ()

commit 3dd2617a7f15dcf08dda4eaae5389ff1e399d222
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Mon Nov 3 17:00:54 2014 +0100

 src/common: Implement roundtoPowOf2 with binary logic

 Instead of using expensive log instructions with floats.
```

C.4 FLASHROM

Flashrom is an utility to read/write flash chips. The contributions of this Thesis to Flashrom extend the project to support SPI EEproms attached to Intel cards. The review process of the contribution was quite tough, as it is the first time that FlashRom adds support for something different than a Flash Chip.

```
commit dc2f6bbc528cefdafead41cd20e075e58a7e3c61
Date:   Mon Jul 28 20:35:21 2014 +0000

 Add new programmer for SPI EEPROMs attached to Intel 82580 NICs.

 This patch lets you read and write the EEPROM on 82580-based gigabit NIC
 cards. So far it has been tested on copper NICs only, but other variants
 employing this controller should work too.
 It is a nice substitution for the official eeupdate tool.

 Speed is quite decent: less than 4 seconds for erases or writes of 32 kB.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Stefan Tauner <stefan.tauner@alumni.tuwien.ac.at>
 Tested-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Acked-by: Stefan Tauner <stefan.tauner@alumni.tuwien.ac.at>
```

C.5 VIDEOLAN CLIENT

VideoLan is a well known multimedia player for many platforms (Linux, Windows, MAC, Android...) and formats (MPEG-2, DivX, H.264, MKV, WebM, WMV)

The contributions fix the support for Video4Linux2 devices with Continuous framesize.

```
commit 4372fe4df714fccb15eeb7e0baeae74fb97edaee
Author: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
Date:   Wed Apr 15 20:42:08 2015 +0200

 v4l2: Fix SetupFormat for CONTINUOUS framesize

 One of the bounds of the loop were wrong, ending up in testing
 formats that were not supported by the hardware.

 Signed-off-by: Ricardo Ribalda Delgado <ricardo.ribalda@gmail.com>
 Signed-off-by: Rémi Denis-Courmont <remi@remlab.net>
```

BIBLIOGRAPHY

[AC06] David M Airlie and Open Source Contractor. Open source graphic drivers-they don't kill kittens. In *Proceedings of the Linux Symposium*, volume 1, pages 19–26. Citeseer, 2006.

[AD514] Anlog devices ad5628 datasheet, 2014.

[AD914a] Analog devices ad9259 datasheet, 2014.

[AD914b] Analog devices ad9970 datasheet, 2014.

[ADBL11] Thomas Arnold, Martin De Biasio, and Raimund Leitner. Hyperspectral video endoscope for intra-surgery tissue classification using auto-fluorescence and reflectance spectroscopy. In *European Conferences on Biomedical Optics*, pages 808711–808711. International Society for Optics and Photonics, 2011.

[ADBLL09] T Arnold, M De Biasio, G Lodron, and R Leitner. Real-time spectral imaging system. In *Proceedings of the IASTED International Conference*, volume 654, page 805, 2009.

[AF99] C. Adams and S. Farrell. RFC2510: Internet X. 509 Public Key Infrastructure Certificate Management Protocols. *Internet RFCs*, 1999.

[Agi80] Gerald J Agin. Computer vision systems for industrial inspection and assembly. *Computer*, (5):11–20, 1980.

[AKM+08] Abdulkareem Alali, Huzefa Kagdi, Jonathan Maletic, et al. What's a typical commit? a characterization of

open source software repositories. In *Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on*, pages 182–191. IEEE, 2008.

[AL10] Diwan P Ariana and Renfu Lu. Hyperspectral imaging for defect detection of pickling cucumbers. *Hyperspectral imaging for food quality analysis and control*, pages 431–447, 2010.

[All] Open Handset Alliance. Camera hal v3 overview.

[AMY09] Shuichi Asano, Tsutomu Maruyama, and Yoshiki Yamaguchi. Performance comparison of fpga, gpu and cpu in image processing. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pages 126–131. IEEE, 2009.

[And08] Erik Andersen. Busybox. *Website], Available HTTP: http://www. busybox. net*, 2008.

[APO14] Kevin Abas, Caio Porto, and Katia Obraczka. Wireless smart camera networks for the surveillance of public spaces. *Computer*, (5):37–44, 2014.

[AR09] Ankit Agrawal and Ramesh Raskar. Optimal single image capture for motion deblurring. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2560–2567. IEEE, 2009.

[AS05] Henrry Andrian and Kai-Tai Song. Embedded cmos imaging system for real-time robotic vision. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1096–1101. IEEE, 2005.

[A/S15] Newtec A/S. How to get the perfect potatoes, July 2015.

[ATP+10] Andrew Adams, Eino-Ville Talvala, Sung Hee Park, David E Jacobs, Boris Ajdin, Natasha Gelfand, Jennifer

Dolson, Daniel Vaquero, Jongmin Baek, Marius Tico, et al. The frankencamera: an experimental platform for computational photography. In *ACM Transactions on Graphics (TOG)*, volume 29, page 29. ACM, 2010.

[avr14] Avr atmega328p datasheet, 2014.

[B+00] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.

[BAGM07] J Blasco, N Aleixos, J Gómez, and E Moltó. Citrus sorting by identification of the most common defects using multispectral computer vision. *Journal of Food Engineering*, 83(3):384–393, 2007.

[Ban13] Deepti Bansal. Comparative study of various systems on chips embedded in mobile devices. *Innovative Systems Design and Engineering*, 4(7):11–19, 2013.

[BAS04] Ravi Budruk, Don Anderson, and Tom Shanley. *PCI express system architecture*. Addison-Wesley Professional, 2004.

[Bay76] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[BB82] Dana H Ballard and Christopher M Brown. Computer vision. *Prenice-Hall, Englewood Cliffs, NJ*, 1982.

[BB14] Merwan Birem and François Berry. Dreamcam: A modular fpga-based smart camera architecture. *Journal of Systems Architecture*, 60(6):519–527, 2014.

[BCJK05] Harry Broers, Wouter Caarls, Pieter Jonker, and Richard Kleihorst. Architecture study for smart cameras. In *Proceedings of the EOS Conference on Industrial Imaging and Machine Vision*, pages 39–49, 2005.

[BD11] Stephen D Burks and Joshua M Doe. Gstreamer as a framework for image processing applications in image fusion. In *SPIE Defense, Security, and Sensing*, pages 80640M–80640M. International Society for Optics and Photonics, 2011.

[BDM+06] Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, and Helmut Schwabach. Distributed embedded smart cameras for surveillance applications. *Computer*, 39(2):68–75, 2006.

[Bha14] Krishnaraj Bhat. clpeak–peak performance of your opencl device. 2014.

[Blo06] Joshua Bloch. How to design a good api and why it matters. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pages 506–507. ACM, 2006.

[BMBJ15] Glenn J Beach, Gary Moody, James Burkowski, and Charles J Jacobus. Portable composable machine vision system for identifying projectiles, March 4 2015. US Patent App. 14/638,313.

[Bol79] Robert C Bolles. Part acquisition using the sri vision module. In *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International*, pages 872–877. IEEE, 1979.

[Boo45] Xilinx Data Book. Bridge for pci express v2.3. *Inc., San Jose, CA*, 20145.

[Boo15] Xilinx Data Book. Axi central direct memory access v4.1. *Inc., San Jose, CA*, 2015.

[Bor09] Anton Borisov. Coreboot at your service! *Linux Journal*, 2009(186):1, 2009.

[Bra] Chris Brady. Memtest86+.

[Bra96] James W Brault. New approach to high-precision fourier transform spectrometer design. *Applied Optics*, 35(16):2891–2896, 1996.

[BRS04] Michael Bramberger, Bernhard Rinner, and Helmut Schwabach. An embedded smart camera on a scalable heterogeneous multi-dsp system. In *Proceedings of the European DSP Education and Research Symposium (EDERS 2004)*, 2004.

[BS04] Tadhg Brosnan and Da-Wen Sun. Improving quality inspection of food products by computer vision—-a review. *Journal of Food Engineering*, 61(1):3–16, 2004.

[CAM+11] Sergio Cubero, Nuria Aleixos, Enrique Moltó, Juan Gómez-Sanchis, and Jose Blasco. Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food and Bioprocess Technology*, 4(4):487–504, 2011.

[cel15] Newtec potato grader: Celox xt p-ms, 2015.

[CF06] Varsha Chikane and Chiou-Shann Fuh. Automatic white balance for digital still cameras. *Journal of Information Science and Engineering*, 22(3):497–509, 2006.

[che15] Newtec qc90sv checkpoint, 2015.

[cmo15] Cmosis cmv family, 2015.

[cog15] Developer's certificate of origin 1.1, August 2015.

[Cor07] Jonathan Corbet. Video4linux2 part 7: Controls. In *LWN*. LWN, 2007.

[Cor11] Jonathan Corbet. The videobuf2 api. In *LWN*. LWN, 2011.

[CRKH05] Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman. *Linux device drivers*. " O'Reilly Media, Inc.", 2005.

[Cut06] Innes C Cuthill. Color perception. *Bird coloration*, 1:3–40, 2006.

[DC+12] ER Davies, DG Caldwell, et al. Machine vision in the food industry. *Robotics and Automation in the Food Industry: Current and Future Technologies*, page 75, 2012.

[Dev13] NumPy Developers. Numpy. *NumPy Numpy. Scipy Developers*, 2013.

[DFRR+10] A Del Fiore, M Reverberi, A Ricelli, F Pinzari, S Serranti, AA Fabbri, G Bonifazi, and C Fanelli. Early detection of toxigenic fungi on maize by hyperspectral imaging analysis. *International Journal of Food Microbiology*, 144(1):64–71, 2010.

[Dir04] Measuring Instruments Directive. Directive 2004/22/ec. *European Council*, 2004.

[DM07] S. Drimer and S.J. Murdoch. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks. *Proceedings of USENIX Security, September*, 2007.

[dM09] Arnaldo Carvalho de Melo. Performance counters on linux. In *Linux Plumbers Conference*, 2009.

[DSAG13] C. Demant, B. Streicher-Abel, and C. Garnica. *Industrial Image Processing: Visual Quality Control in Manufacturing*. Professional computing. Springer Berlin Heidelberg, 2013.

[DVR99] Bill Dirks, Hans Verkuil, and Martin Rubli. Video for linux two api specification. *History*, 6:11, 1999.

[DWL+12] Peng Du, Rick Weber, Piotr Luszczek, Stanimire Tomov, Gregory Peterson, and Jack Dongarra. From cuda to opencl: Towards a performance-portable solution for multi-platform gpu programming. *Parallel Computing*, 38(8):391–407, 2012.

[dyn15] Device tree dynamic resolver notes, August 2015.

[FAO09] Food FAOstat. agriculture organization of the united nations. *Statistical database*, 2009.

[Fei15] Werner Feith. Pro and con of using gen¡ i¿ cam based standard interfaces (gev, u3v, cxp and clhs) in a camera or image processing design. In *IS&T/SPIE Electronic Imaging*, pages 940511–940511. International Society for Optics and Photonics, 2015.

[Fil03] Andry Filippov. Reconfigurable high resolution network camera. In *Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on*, pages 276–277. IEEE, 2003.

[fin] Fintek f81216a datasheet.

[fla15] Flashrom, August 2015.

[fpa15] Andata fpa320x256-c datasheet, 2015.

[FS05] S. Fleck and W. Strasser. Adaptive Probabilistic Tracking Embedded in a Smart Camera. *Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on*, 3, 2005.

[FVG01] Filip Feyaerts and Luc Van Gool. Multi-spectral vision system for weed detection. *Pattern Recognition Letters*, 22(6):667–674, 2001.

[FVMS+04] Felix Friedl-Vallon, Guido Maucher, Meinhard Seefeldner, Olaf Trieschmann, Anne Kleinert, Anton Lengel,

Corneli Keim, Hermann Oelhaf, and Herbert Fischer. Design and characterization of the balloon-borne michelson interferometer for passive atmospheric sounding (mipas-b2). *Applied optics*, 43(16):3335–3355, 2004.

[FZWL12] Xianghua Fan, Fuyou Zhang, Haixia Wang, and Xiao Lu. The system of face detection based on opencv. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 648–651. IEEE, 2012.

[Gai03] J. Gaisler. The LEON-2 Processor User's Manual. *Gaisler Research, November*, 2003.

[Gas14] Harris Gasparakis. Heterogeneous compute in computer vision: Opencl in opencv. In *IS&T/SPIE Electronic Imaging*, pages 90290L–90290L. International Society for Optics and Photonics, 2014.

[GFOG+07] J. Galbally, J. Fierrez, J. Ortega-Garcia, M. R. Freire, F. Alonso-Fernandez, J. A. Siguenza, J. Garrido-Salas, E. Anguiano-Rey, G. Gonzalez de Rivera, R. Ribalda, M. Faundez-Zanuy, J. A. Ortega, V. Cardeñoso-Payo, A. Viloria, C. E. Vivaracho, Q. I. Moro, J. J. Igarza, J. Sanchez, I. Hernaez, and C. Orrite-Uruñuela. Biosecurid: a multimodal biometric database. In *Proc. MADRINET Workshop*, pages 68–76, November 2007.

[GGKP08] Scott Grauer-Gray, Chandra Kambhamettu, and Kannappan Palaniappan. Gpu implementation of belief propagation using cuda for cloud tracking and reconstruction. In *IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS 2008)*, volume 4, page 2, 2008.

[GR03] Kai Germaschewski and Sam Ravnborg. Kernel configuration and building in linux 2.5. In *Linux Symposium*, page 185, 2003.

[Gui11] Erico Guizzo. How google's self-driving car works. *IEEE Spectrum Online, October*, 18, 2011.

[Gunoo] Sundaram Gunasekaran. *Nondestructive Food Evaluation: Techniques to Analyze Properties and Quality*. CRC Press, 2000.

[GWMW01] MD Garris, CI Watson, RM McCabe, and CL Wilson. User's Guide to NIST Fingerprint Image Software(NFIS). *2001.*, 2001.

[GZ03] Á. Gutiérrez and D. Zurdo. *Comercio electrónico y privacidad en Internet: Derechos y deberes en la sociedad de la información:(real Decreto 34-2002, de servicios de la sociedad de la información y de comercio electrónico)*. Creaciones Copyright, 2003.

[HD07] John Hunter and Darren Dale. The matplotlib user's guide. *Matplotlib user's guide*, 2007.

[HE05] S. Heithecker and R. Ernst. Traffic shaping for an FPGA based SDRAM controller with complex QoS requirements. *Proceedings of the 42nd annual conference on Design automation*, pages 575–578, 2005.

[Her15] Esko Herrala. Imaging spectrometer, March 31 2015. US Patent 8,994,939.

[HKLP09] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1430–1437. IEEE, 2009.

[HRJ+05] Glenn D Hines, Zia-ur Rahman, Daniel J Jobson, Glenn A Woodell, and Steven D Harrah. Real-time enhanced vision system. In *Defense and Security*, pages 127–134. International Society for Optics and Photonics, 2005.

[HSH+13] Michael Hahnle, Frerk Saxen, Matthias Hisung, Ulrich Brunsmann, and Konrad Doll. Fpga-based real-time pedestrian detection on high-resolution images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 629–635. IEEE, 2013.

[Hyn] SK Hynix. Teardown xbox one. *Engineering Technology*.

[ICX] Sony icx204al datasheet.

[ime15] Imec hyperspectral imaging, 2015.

[Inc13] AMD Inc. G-series platform brief, June 2013.

[ISK01] Y. Isobe, Y. Seto, and M. Kataoka. Development of personal authentication system using fingerprint with digital signature technologies. *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 9, 2001.

[JA10] Johannes Jordan and Elli Angelopoulou. Gerbil-a novel software framework for visualization and analysis in the multispectral domain. 2010.

[jac15] jackshencn. Peeping into pixel – a micrograph of cmos sensor, July 2015.

[Jan12] Juhani Janhunen. Quality inspection of a production line with a smart camera. 2012.

[JB04] Dirk Jansen and Holger Buttner. Real-time ethernet: the ethercat solution. *Computing and Control Engineering*, 15(1):16–21, 2004.

[JCD07] Anil K Jain, Yi Chen, and Meltem Demirkus. Pores and ridges: High-resolution fingerprint matching using level 3 features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):15–27, 2007.

[JCP+10] Seunghun Jin, Junguk Cho, Xuan Dai Pham, Kyoung Mu Lee, S-K Park, Munsang Kim, and Jae Wook Jeon. Fpga design and implementation of a real-time stereo vision system. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(1):15–26, 2010.

[JGB04] C. T. Johnston, K. T. Gribbon, and D. G. Bailey. Implementing image processing algorithms on FPGAs. In *ENZCON '04: Proceedings of the 11th Electronics New Zealand conference*, pages 118–123, 2004.

[JK03] J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS)# 1: RSA Cryptography Specifications Version 2.1. Technical report, RFC 3447, February 2003, 2003.

[JNN08] A.K. Jain, K. Nandakumar, and A. Nagar. Biometric Template Security. *Journals on Advances in Signal Processing, Hindawi Publishing Corporation*, 2008.

[JOP14] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: Open source scientific tools for {Python}. 2014.

[KA12] Tamotsu Koiwai and Hidenori Akita. Development of compact lenses for micro four-thirds system. *Journal of The Society of Photographic Science and Technology of Japan*, 74:56–62, 2012.

[KCM08] Greg KroahHartman, Jonathan Corbet, and Amanda McPherson. Linux kernel development. *the Linux foundation*, 2008.

[KGWK96] Tina Kapur, W Eric L Grimson, William M Wells, and Ron Kikinis. Segmentation of brain tissue from magnetic resonance images. *Medical image analysis*, 1(2):109–127, 1996.

[kin15] Xilinx kintex-7 fpga user-guide, 2015.

[KK79] Ilkka Kuorinka and Paavo Koskinen. Occupational rheumatic diseases and upper limb strain in manual jobs in a light mechanical industry. *Scandinavian journal of work, environment & health*, pages 39–47, 1979.

[KL08] John A Kalomiros and John Lygouras. Design and evaluation of a hardware/software fpga-based system for fast image processing. *Microprocessors and Microsystems*, 32(2):95–106, 2008.

[KLB+13] Mads RB Kristensen, Simon AF Lund, Troels Blum, Kenneth Skovhede, and Brian Vinter. Bohrium: unmodified numpy code on cpu, gpu, and cluster. *Python for High Performance and Scientific Computing (PyHPC 2013)*, 2013.

[KMW+06] J. Koreman, AC Morris, D. Wu, S. Jassim, H. Sellahewa, J. Ehlers, G. Chollet, G. Aversano, H. Bredin, S. Garcia-Salicetti, et al. Multi-modal biometric authentication on the SecurePhone PDA. *Multimodal User Authentication, To appear in Proc. MMUA*, 2006.

[KN12] Jaromír Krpec and Martin Nemec. Face detection cuda accelerating. 2012.

[Kos79] Hiroyasu Koshimizu. Fundamental study on automatic fabric inspection by computer image processing. In *Techincal Symposium East*, pages 30–37. International Society for Optics and Photonics, 1979.

[kpa15] How to get your change into the linux kernel, August 2015.

[kps15] Kernel patch statistic, August 2015.

[Kum08] Ajay Kumar. Computer-vision-based fabric defect detection: a survey. *Industrial Electronics, IEEE Transactions on*, 55(1):348–363, 2008.

[Kö] Marianne König. Creating meteorological products from satellite data. *Eutmetsat*.

[LBo8] Grant Likely and Josh Boyer. A symphony of flavours: Using the device tree to describe embedded hardware. In *Proceedings of the Linux Symposium*, volume 2, pages 27–37, 2008.

[LEWMo5] K Lee, John C Eidson, Hans Weibel, and Dirk Mohl. Ieee 1588-standard for a precision clock synchronization protocol for networked measurement and control systems. In *Conference on IEEE*, volume 1588, page 2, 2005.

[LFGo6] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *Information Forensics and Security, IEEE Transactions on*, 1(2):205–214, 2006.

[LKo1] Suk-Han Lam and Chi-Wan Kok. Demosaic: Color filter array interpolation for digital cameras. In *Advances in Multimedia Information Processing—PCM 2001*, pages 1084–1089. Springer, 2001.

[LLKKo5] C. Lee, S. Lee, J. Kim, and S.J. Kim. Preprocessing of a Fingerprint Image Captured with a Mobile Camera. *Advances in Biometrics: International Conference*, 2005.

[LMo1] E Scott Larsen and David McAllister. Fast matrix multiplies using graphics hardware. In *Proceedings of the 2001 ACM/IEEE conference on Supercomputing*, pages 55–55. ACM, 2001.

[lm9] Texas instruments lm92 datasheet.

[LPS+10] Louis Longchamps, Bernard Panneton, Guy Samson, Gilles D Leroux, and Roger Thériault. Discrimination of corn, grasses and dicot weeds by their uv-induced fluorescence spectral signature. *Precision agriculture*, 11(2):181–197, 2010.

[LSH13] Arttu Leppakoski, Erno Salminen, and Timo D Hamalainen. Framework for industrial embedded system product development and management. In *System on Chip (SoC), 2013 International Symposium on*, pages 1–6. IEEE, 2013.

[LT15] Michael Larabel and M Tippett. Phoronix, 2015.

[LTO05] N. Lepisto, B. Thornberg, and M. O'Nils. High-performance FPGA based camera architecture for range imaging. *Proceedings of the 23rd NORCHIP Conference*, pages 165–168, 2005.

[MA04] F Mendoza and JM Aguilera. Application of image analysis for classification of ripening bananas. *Journal of food science*, 69(9):E471–E477, 2004.

[Mag] EandT Magacine. *Engineering Technology*.

[MC07] Eric Monmasson and Marcian N Cirstea. Fpga design methodology for industrial control systems—a review. *Industrial Electronics, IEEE Transactions on*, 54(4):1824–1842, 2007.

[MGMEPP+11] Rafael J Montero-Gonzalez, Arturo Morgado-Estevez, Fernando Perez-Peña, Alejandro Linares-Barranco, Angel Jimenez-Fernandez, Bernabe Linares-Barranco, and Jose Antonio Perez-Carrasco. Visual aer-based processing with convolutions for a parallel supercomputer. In *Signal Processing and Multimedia Applications (SIGMAP), 2011 Proceedings of the International Conference on*, pages 1–6. IEEE, 2011.

[MGMG11] Aaftab Munshi, Benedict Gaster, Timothy G Mattson, and Dan Ginsburg. *OpenCL programming guide*. Pearson Education, 2011.

[MGPC03] D. Moon, Y.H. Gil, S.B. Pan, and Y. Chung. Implementation of the USB Token System for Fingerprint Verification. *Proc. of Scandinavian Conf. on Image Analysis (LNCS 2749)*, pages 998–1005, 2003.

[MGS98] Anil Mital, M Govindaraju, and B Subramani. A comparison between manual and hybrid methods in parts inspection. *Integrated Manufacturing Systems*, 9(6):344–349, 1998.

[Mil91] David L Mills. Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10):1482–1493, 1991.

[MKS+06] AC Morris, J. Koreman, H. Sellahewa, J. Ehlers, S. Jassim, L. Allano, and S. Garcia-Salicetti. The SecurePhone PDA database, experimental protocol and automatic test procedure for multimodal user authentication. Technical report, Tech Report, Jan. 2006., 2006.

[MM10] Erich Marth and Guillermo Marcus. Parallelization of the x264 encoder using opencl. In *ACM SIGGRAPH 2010 Posters*, page 72. ACM, 2010.

[MMJ+07] Larry Matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas, Andrew Stein, and Anelia Angelova. Computer vision on mars. *International Journal of Computer Vision*, 75(1):67–92, 2007.

[Moc02] Patrick Mochel. The linux kernel device model. In *Ottawa Linux Symposium*, page 368, 2002.

[Moc05] Patrick Mochel. The sysfs filesystem. In *Linux Symposium*, page 313, 2005.

[Mur] Janet Murray. Modularity and project ara.

[MV96] James D Murray and William VanRyper. Encyclopedia of graphics file formats. *Sebastopol: O'Reilly,— c1996, 2nd ed.*, 1, 1996.

[ND10] John Nickolls and William J Dally. The gpu computing era. *IEEE micro*, (2):56–69, 2010.

[New] Newtec. Celox xt optical potato grader.

[Nie07] KS Niel. Industrial computer vision. *Proceedings of Image and Vision Computing*, pages 198–204, 2007.

[NSK+05] Karl O Nakken, Marit H Solaas, Marianne J Kjeldsen, Mogens L Friis, John M Pellock, and Linda A Corey. Which seizure-precipitating factors do patients with epilepsy most frequently report? *Epilepsy & Behavior*, 6(1):85–89, 2005.

[p1612] Microchip pic16f1823 datasheet, 2012.

[p1811] Microchip p18f87k22 datasheet, 2011.

[Pau12] Lewes F Pau. *Computer vision for electronics manufacturing*. Springer Science & Business Media, 2012.

[PC09] Giuseppe Parziale and Yi Chen. Advanced technologies for touchless fingerprint recognition. In *Handbook of Remote Biometrics*, pages 83–109. Springer, 2009.

[PCM79] GB Porter, TM Cipolla, and JL Mundy. Automatic visual inspection of blind holes in metal surfaces. In *Proc. 1979 Conf. Pattern Recognition and Image Processing*, pages 83–86, 1979.

[Pea09] Tom Pearson. Hardware-based image processing for high-speed inspection of grains. *Computers and electronics in agriculture*, 69(1):12–18, 2009.

[PG07] Fernando Pérez and Brian E Granger. Ipython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, 2007.

[plx] Avago usb3380 datasheet.

[Rec02] ITURBT Recommendation. 709. *Parameter values for the HDTV standards for production and international programme exchange*, 709, 2002.

[Rec11] ITURBT Recommendation. 601. *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios*, 601, 2011.

[Rib13] et. al. Ribalda, Ricardo. Notes on the media summit 2013. In *Kernel Summit*. Linux Foundation, 2013.

[Rib14] et. al. Ribalda, Ricardo. Media summit report 2014. In *Linux Plumbers*. Linux Foundation, 2014.

[RRN02] A Rowe, C Rosenberg, and I Nourbakhsh. Cmucam: a lowoverhead vision system. In *Proceedings, IROS 2002*, 2002.

[RS06] Eraldo Ribeiro and Mubarak Shah. Computer vision for nanoscale imaging. *Machine Vision and Applications*, 17(3):147–162, 2006.

[RTGM03] Piyush Kumar Rai, Kamal Tiwari, Prithwijit Guha, and Amitabha Mukerjee. A cost-effective multiple camera vision system using firewire cameras and software synchronization. In *Proceedings of the 10th International Conference on High Performance Computing, Hyderabad, India*, volume 1720, 2003.

[RVDCJG+08] Ángel Rodríguez-Vázquez, Rafael Domínguez-Castro, Francisco Jiménez-Garrido, Sergio Morillas, Juan Listán, Luis Alba, Cayetana Utrera, Servando Espejo, and

Rafael Romay. The eye-ris cmos vision system. In *Analog circuit design*, pages 15–32. Springer, 2008.

[SA14] Otavio Salvador and Daiane Angolini. *Embedded Linux Development with Yocto Project*. Packt Publishing Ltd, 2014.

[SAW+08] Mohammed A Salem, Markus Appel, Frank Winkler, Beate Meffert, et al. Fpga-based smart camera for 3d wavelet-based image segmentation. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–8. IEEE, 2008.

[SB00] Robert L Shewfelt and Bernhard Bruckner. *Fruit and vegetable quality: an integrated view*. CRC Press, 2000.

[SB09] Mukul Shirvaikar and Tariq Bushnaq. A comparison between dsp and fpga platforms for real-time imaging applications. In *IS&T/SPIE Electronic Imaging*, pages 724406–724406. International Society for Optics and Photonics, 2009.

[SC13] Edward H Shortliffe and James J Cimino. *Biomedical informatics: computer applications in health care and biomedicine*. Springer Science & Business Media, 2013.

[sco15] Newtec scout, 2015.

[SD01] C Setchell and EL Dagless. Vision-based road-traffic monitoring sensor. *IEE Proceedings-Vision, Image and Signal Processing*, 148(1):78–84, 2001.

[sen15] Sentisight sdk, August 2015.

[SH06a] K. Shimizu and S. Hirai. CMOS+ FPGA Vision System for Visual Feedback of Mechanical Systems. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2060–2065, 2006.

[SHo6b] Kazuhiro Shimizu and Shinichi Hirai. Cmos+ fpga vision system for visual feedback of mechanical systems. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2060–2065. IEEE, 2006.

[Sin14] Jaspinder Pal Singh. Designing an fpga synthesizable computer vision algorithm to detect the greening of potatoes. *arXiv preprint arXiv:1403.1974*, 2014.

[Sla90] P Slanina. Solanine (glycoalkaloids) in potatoes: toxicological evaluation. *Food and Chemical Toxicology*, 28(11):759–761, 1990.

[Sol12] Jan Erik Solem. *Programming Computer Vision with Python: Tools and algorithms for analyzing images*. " O'Reilly Media, Inc.", 2012.

[spa15] Xilinx spartan-6 fpga user-guide, 2015.

[ST80] J.S. Sukonick and G.J. Tilden. Method for dynamically viewing image elements stored in a random access memory array, April 8 1980. US Patent 4,197,590.

[ST07a] D. Searls and J. Thompson. The ultimate Linux handheld. *Linux Journal*, 2007(161), 2007.

[ST07b] Yu Shi and Timothy Tsui. An fpga-based smart camera for gesture recognition in hci applications. In *Computer Vision–ACCV 2007*, pages 718–727. Springer, 2007.

[Sta11] AMD Staff. Opencl^{TM} and the amd app sdk v2. 4, 2011.

[Ste79] Warren M Sterling. Automatic non-reference inspection of printed wiring boards. In *Proceeding IEEE Computer Society Conference Pattern Recognition and Image Processing*, pages 93–100, 1979.

[Ste11] Ashley Stevens. Introduction to amba 4 ace. *ARM whitepaper, June*, 2011.

[Sun00] Da-Wen Sun. Inspecting pizza topping percentage and distribution by a computer vision method. *Journal of food engineering*, 44(4):245–249, 2000.

[SWC04] Chris Sullivan, Alex Wilson, and Stephen Chappell. Using c based logic synthesis to bridge the productivity gap. In *Proceedings of the 2004 Asia and South Pacific Design Automation Conference*, pages 349–354. IEEE Press, 2004.

[SYJ14] Chun Lei Shi, Dan Dan Yang, and Guang Yuan Jiang. Research of adaboost face detection and opencv implementation. In *Applied Mechanics and Materials*, volume 651, pages 482–485. Trans Tech Publ, 2014.

[TAK06] Jason G Tong, Ian DL Anderson, and Mohammed AS Khalid. Soft-core processors for embedded systems. In *Microelectronics, 2006. ICM'06. International Conference on*, pages 170–173. IEEE, 2006.

[TBW+01] Wim Taymans, Steve Baker, Andy Wingo, R Bultje, and Stefan Kost. Gstreamer application development manual, 2001.

[TCYT12] Xunqiang Tao, Xinjian Chen, Xin Yang, and Jie Tian. Fingerprint recognition with identical twin fingerprints. *PloS one*, 7(4):61, 2012.

[tin15] Tiny pic bootloader, August 2015.

[TNI+10] Ryoji Tsuchiyama, Takashi Nakamura, Takuro Iizuka, Akihiro Asahara, Satoshi Miki, and Satoru Tagawa. The opencl programming book. *Fixstars Corporation*, 63, 2010.

[Tod02] B.T. Today. Smartcards use biometrics. *Biometric Technology Today*, 10(5):9–11, 2002.

[Tor] Linus Torvalds. Kernel home page. http://kernel.org.

[uli14] Ulis ul 05 25 1-026 datasheet, 2014.

[vdBW05] M. van den Braak and S. Wong. FPGA implementation of Voice-over IP. *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing*, pages 338–342, 2005.

[Vec06] VectorInternational. Bci4-6600 datasheet. http://www.vector-international.be/C-Cam/doc/Bci4-6600.pdf, 2006.

[Ver11] Geert Verhoeven. Taking computer vision aloft–archaeological three-dimensional reconstructions from aerial photographs with photoscan. *Archaeological Prospection*, 18(1):67–73, 2011.

[Ver12] HALCON Version. 11.0. 1–halcon/hdevelop reference manual. *MVTec Software GmbH*, 2012.

[Vis13] Smarter Vision. Xcelljournal xcelljournal. 2013.

[VMC02] J. Viega, M. Messier, and P. Chandra. *Network Security with OpenSSL*. O'Reilly, 2002.

[web15] Newtec web services, 2015.

[WG70] B. W and S. G. Three dimensional charge coupled devices, March 12 1970. US Patent 3,796,927.

[Wil] D Wilson. The [almost definitive] fourcc definition list, rgb/yuv conversion.

[WK11] Desheng Wang and Lingzhou Kang. Image capture and preprocessing system based on fpga. *Dianshi Jishu(Video Engineering)*, 35(3):32–35, 2011.

[WOG+12] W Woiwode, H Oelhaf, T Gulde, C Piesch, G Maucher, A Ebersoldt, C Keim, M Höpfner, S Khaykin, F Ravegnani, et al. Mipas-str measurements in the arctic utls in winter/spring 2010: instrument characterization, retrieval and validation. *Atmospheric measurement techniques*, 5(6):1205–1228, 2012.

[WVH97] John Woodfill and Brian Von Herzen. Real-time stereo vision on the parts reconfigurable computer. In *Field-Programmable Custom Computing Machines, 1997. Proceedings., The 5th Annual IEEE Symposium on*, pages 201–210. IEEE, 1997.

[xc315] Xc3sprog, 2015.

[Xil06] IARR Xilinx. Microblaze processor reference guide. *reference manual*, 23, 2006.

[YMB+03] Jean-Jacques Yon, E Mottin, L Biancardini, L Letellier, and JL Tissot. Infrared microbolometer sensors and their application in automotive safety. In *Advanced Microsystems for Automotive Applications 2003*, pages 137–157. Springer, 2003.

[YY15] Vipin Yadav and Vishal Yadav. Challenging and oppourtunities of project ara. *IT INTELLIGENCE INNOVATIONS-2015*, page 1, 2015.

[YZD08] Cheng-hua YAN, Yu ZHOU, and Si-dan DU. Rs485 communication protocol based on embedded linux. *Computer Engineering*, 11:101, 2008.

[ZPF97] Zhimin Zhou, Bedabrata Pain, and Eric R Fossum. Frame-transfer cmos active pixel sensor with pixel binning. *IEEE Transactions on electron devices*, 44(10):1764–1768, 1997.

[ZWO+11] Daniel Ziener, Stefan Wildermann, Andreas Oetken, Andreas Weichslgartner, and Jürgen Teich. A flexible smart camera system based on a partially reconfigurable dynamic fpga-soc. In *Proceedings of the Workshop on Computer Vision on Low-Power Reconfigurable Architectures at the FPL*, volume 2011, pages 29–30, 2011.

EPILOGUE

Yo no soy bueno para estas hóspedas, pero... que gran día para mí!. Se casa la chiquilla: la hija de Jose Mari y Aurori con mi chaval: el mayor, abogado, recién sacada la carrera, con ~~30~~ 32 años... échale huevos.

Serafín (Karlos Arguiñano), Airbag

Science is a way of life. Science is a perspective. Science is the process that takes us from confusion to understanding in a manner that's precise, predictive and reliable - a transformation, for those lucky enough to experience it, that is empowering and emotional.

Brian Greene